ファミリーコンピュータ™
ファミリーベーシック™

絵でわかるファミコンベーシック

新星出版社
385

ファミリー コンピュータ™
ファミリー ベーシック™

絵でわかる ファミコンベーシック

山下利秋

新星出版社

地上冒険マップつき
ゼルダの伝説
必勝攻略ガイド
ひみつをあかすヒントがぎっしり

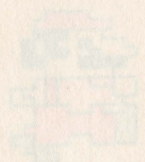
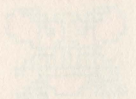
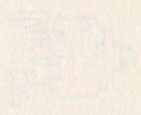
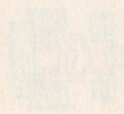
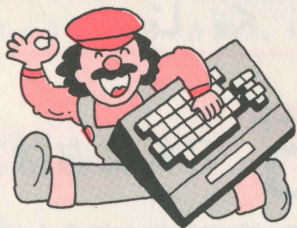
任天堂 ファミリーコンピュータ™

ファミコンゲーム必勝攻略法
スーパーマリオ／ドルアーガの塔
スターフォース／スパルタンX
逆転Vテクきみがチャンピオン！

絵でわかる

ファミコンベーシック

山下利秋

新星出版社

はじめに

きみ、ファミコンしちゃってる？

たくさんたくさん、ファミリーコンピュータゲームで楽しんでいるんだろうね。長い時間遊びすぎて、おとうさんやおかあさんにしかられたりしなかったかな？

ただただ、ファミコンゲームでピコピコと遊ぶだけじゃなくて、ファミコンのキーボードを手に入れたのだから、こんどはキーボードを使って遊ぶって方法もあるよ。でも、むずかしくって、つまらないかい？

はじめてキーボードに向かって、初歩のベーシックからひとつずつ進んでいくのもいいけれど、楽しみながら、遊びながらプログラミングをおぼえられたら、こんないいことはないだろう。

この本は、楽しく遊びながら、ファミコンのベーシックをなんとなくおぼえてしまうようなねらいで書かれているんだ。いろいろなプログラムがのっていて、いろいろな説明も書いてあるけれど、めんどうだったら説明には目をつぶって、プログラムだけをどんどんキー入力してRUNさせれば、それだけで楽しいしおもしろいし、知らないうちにプログラミングもおぼえてしまうはずだ。がんばって遊んでね。

もくじ

パート0　こんにちはファミコンベーシック

パート1　マリオ、画面に出てこい

パート2 アニメキャラといっしょ

パート3 キャラクタと自由に遊ぼう

4 BGグラフィックで絵を描いて

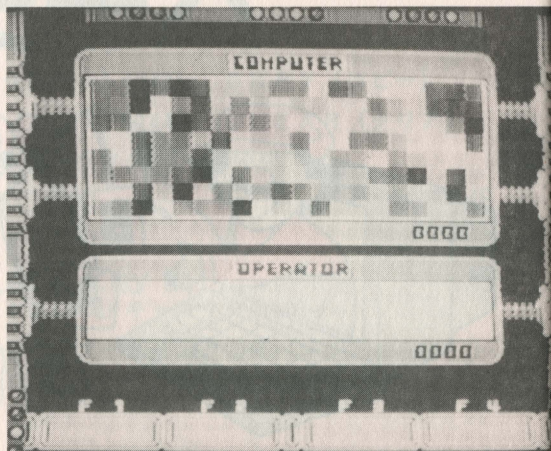
パート1

こんにちは
ファミコン
ベーシック

ファミコンになにをさせる？

♥計算する？　作曲する？ それとも、うらないする？

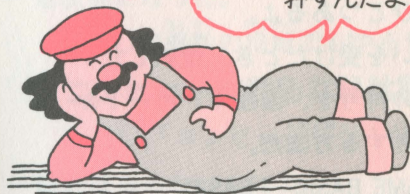
ファミコンのキーボードを手に入れたきみは、きょうから一流のパソコニストの道をまっしぐらにGo! だ。

ファミリーベーシックのカートリッジを本体に接続したら、まず電源ON。テレビの画面にコンピュータの絵が出てくるぞ。これをスタート画面っていうんだ。どれでもいいから、キーを押してみよう。画面が変わって、ほら、ファミコンがきみに話しかけてくるよ。話しかけられたのだから、それに答えてあげよう。

ファミコンと画面で会話を始めてみよう

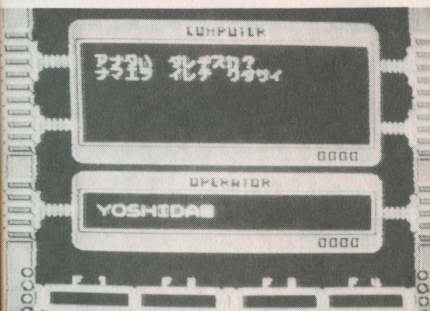
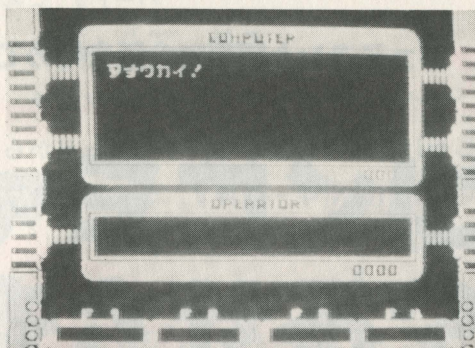
「アナタハ　ダレデスカ？　ナマエヲ　イレテクダサイ」って話しかけてきたのだから、キーボードから自分の名まえを入れてから RETURN キーを押す。名まえは、ローマ字でもカタカナでもいいよ。

きみの名まえは？　ン？　マリオだって？　それじゃ、「マリオ」と入れてみよう。

名まえがめんどうだったらなにもしないで RETURN だけでもいいんだ

カタカナは カナ キーを押してからその文字のキーを押すんだよ

＊ファミリーベーシックになにができる？

ファミリーベーシックには、ベーシックのほかに、次の5つのソフトが入っているんだよ。それぞれのソフトは、いったいなにをするんだろう？

1．GAME BASIC…ゲームのプログラム作り。
2．カリキュレータボード…いろいろな計算をする。
3．ミュージックボード…作曲をする。
4．メッセージボード…伝言板や記録板の役目をする。
5．コンピュータうらない…コンピュータうらないをする。

そう、きみはまさしくマリオなのだ。ファミコンは、ちゃんと「リョウカイ」したではないか。つづいて、ファミコンからのメッセージ。

●ハイかイイエと答えたら RETURN だ

GAME BASICをしたい？　したかったら「ハイ」のキー、したくなかったら「イイエ」のキーを押すんだ。

「ハイ」のキーは F1 キー
「イイエ」のキーは F2 キーだよ

＊機能を選ぶもうひとつの方法

ファミコンベーシックでなにをするかを選択するとき、画面の「〜ニ　シマスカ？」の問いに「ハイ」「イイエ」で答えるほかに方法がもうひとつあるよ。

問いを受けたとき、選択したい機能の符号を、直接キーボードから入力する方法だ。

- GAME BASIC → BASIC
- カリキュレータボード → CAL.
- ミュージックボード → MUS.
- メッセージボード → MES.

たとえば、ミュージックボードにしたかったら「MUS.」と入力して RETURN だ。

「イイエ」のキーを押せば、ファミコンは次のメッセージを出して聞いてくるから、自分がしたいものが出てきたときに、「ハイ」と答えればいい。

では、カリキュレータボードを動かしてみようか。

「"GAME BASIC" ニ　シマスカ」の答えは F 2 キー（イイエ）だね。そして RETURN。

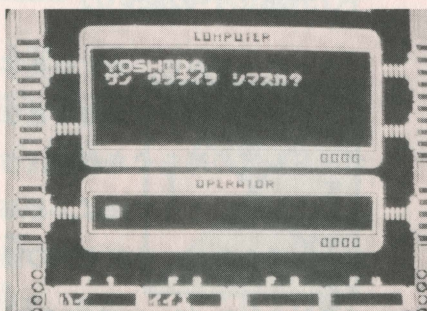
つづいて「"カリキュレータボード" ニ　シマスカ」と画面に出るから、こんどは F 1 キー（ハイ）だよ。そして RETURN キーだ。

これでファミコンは、どんな計算でもできる状態になった。

ためしに算数の宿題でもやらせてみよう。

●コンピュータうらないは別の方法で

ただし、「コンピュータうらない」だけは、別の方法で起動しなければならないんだよ。

「～ニ　シマスカ？」と画面に出たら、ファミコンにあいさつしてみよう。あいさつのことばは、「HELLO」「オハヨウ」「コンニチハ」「コンバンハ」のどれでもいい。もちろん、キーボードから、このことばを入力して

やらなければ、ファミコンには聞こえない。そして、RETURN キーだ。

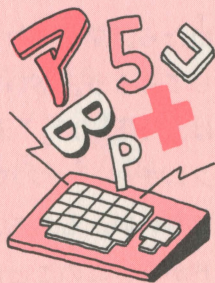
♥ベーシックをクイックスタートさせてみよう

ベーシックの起動は、画面が「"GAME BASIC" ニ　シマスカ？」とたずねてきたとき、「ハイ」RETURN キーとすればいい、といったけど、もっと簡単にする方法がある。この方法をクイックスタートというんだ。

電源を入れたあと、T キーを押しながら、ファミコンのリセットスイッチを押せばいい。これで、GAME BASIC モードに移れる。

キーボードの使い方

キーボードから、カナや英字、数字、記号などを、思い通りに入力できなければ、キーボードも宝の持ちぐされだ。

ここでは、どうすれば、どんな文字や記号が入力できるか、そのキーの使い方をいちおう書いておこう。

●ベーシックとBG(ビージー)グラフィックスタート

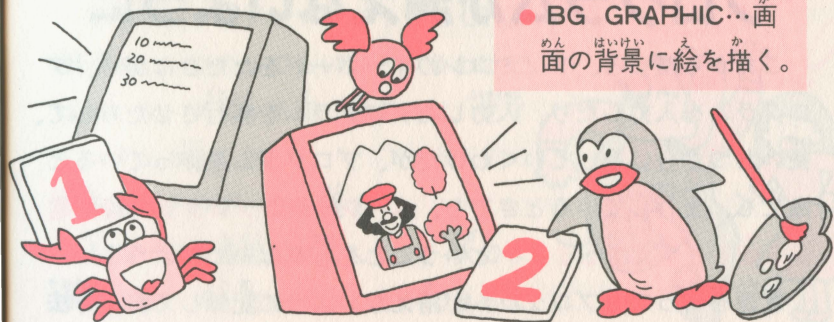
GAME　BASICモードには、BASICとBG(ビージー) GRAPHIC(グラフイック)の2つの機能(きのう)があるから、次(つぎ)にはこのどちらを使(つか)うかを、ファミコンに教(おし)えてやるというのが順序(じゅんじょ)だよ。

キーボードの[1]キーを押(お)せば、BASICがスタートだ。

キーボードの[2]キーを押(お)せば、BG　GRAPHICがスタートだ。

＊BASICとBG GRAPHIC

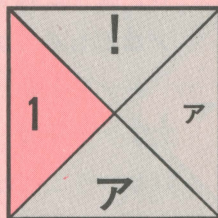
GAME　BASICモードには2つの機能(きのう)が入(はい)っている。

- BASIC…ベーシックでプログラムを入(にゅう)力(りょく)したり、プログラムを実行(じっこう)したりする。
- BG　GRAPHIC…画(が)面(めん)の背景(はいけい)に絵(え)を描(か)く。

キーボードの使(つか)い方(かた)

文字(もじ)キーの使(つか)い方(かた)

カタカナ、英字(えいじ)、数字(すうじ)、記号(きごう)を入力(にゅうりょく)するのが文字(もじ)キーだ。ひとつの文字(もじ)キーで2種類(しゅるい)から4種類(しゅるい)の文字(もじ)や記号(きごう)が入力(にゅうりょく)できるから、まず、その使(つか)い分(わ)けをはっきりおぼえておこう。

●**英字(えいじ)と数字(すうじ)**……キーをそのまま押(お)す。ただ

ステップ　2

プログラムのセーブとロード

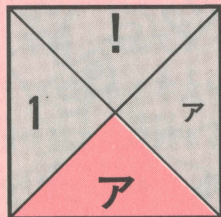
♥電源スイッチを切ってもプログラムが消えないように

これからずっと、ファミコンのキーボードをたたきながら、プログラムを入力したり、入力したプログラムを実行させたりして、長いおつき合いをしていくわけだが、プログラムを作っているときでも、実行しているときでも、**「ベーシックのプログラムは、電源を切ると消えてしまう」**ということを忘れないでいよう！

電源を切ってもプログラムが消えないように記録しておく方法

キーボードの使い方

し、英字と数字以外に、￥＠〔；：〕，／なども そのまま押すだけでいい。

キーを４等分して、左側に書いてある文字や記号は、ただキーを押すだけでいいということだ。

を、プログラムのセーブというよ。

プログラムは、カセットレコーダの中にセーブする方法と、メモリバックアップ機能を生かして、一時的にプログラムを記憶させておく方法がある。

●メモリのバックアップ

ファミリーベーシックのカセットは、バックアップスイッチをONにしておけば、ファミリーコンピュータの電源を切っても、メモリにあるデータを記憶させておくことができるんだ。

操作中やたらにON、OFFするとヘンな動作をすることがあるよ

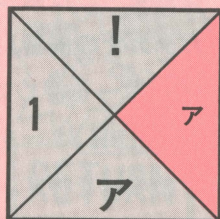
でも、バックアップスイッチのON、OFFは、いつやってもいいというわけではないよ。スイッチ切り替えには、タイミングがあるんだ。

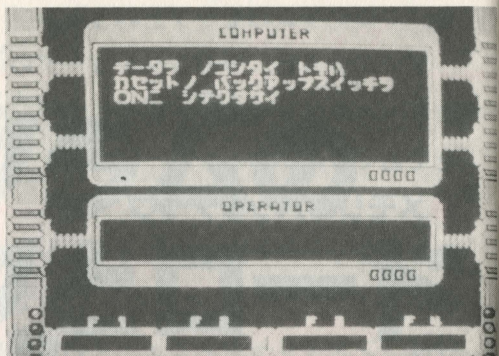
■各ボードのデータを残す

各ボードの使用が終了したら、次のようにして、データを残すことができるよ。

キーボードの使い方

● **カタカナ**……［カナ］キーを押してから、使いたいカタカナの書いてあるキーを押す。（キーの下側に書いてある文字）

● **カタカナ小文字**……［カナ］キーを押してカナモードにしてから、［SHIFT］キーを押しながら、使いたい字のキーを押す。

(1) ESC キーを押して、セレクト画面に戻す。

(2) 画面で「～シマスカ」と聞いてきたら、オ ワ リ ⏎ または F 4 ⏎ とする。

画面に「データヲ ノコシタイトキハ カセットノバックアップスイッチヲ ONニシテクダサイ」と表示される。

(3) バックアップスイッチをONにする。

このあとは、カセットを本体から抜いても、データは記憶されているよ。

■抜いたカセットを再び本体に接続したら

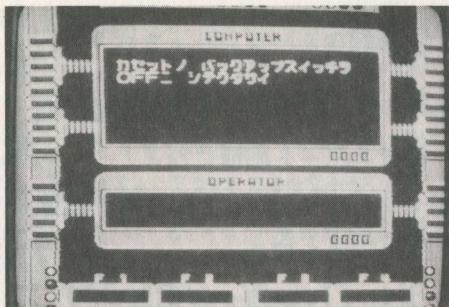
バックアップスイッチをONにして、本体からカセットを引き抜いたあと、もう一度、カセットを差し込んで、操作をつづけるときは，次のようにしよう。

キーボードの使い方

（キーの右側に書いてある文字）

●**カタカナの濁音（ガギグゲゴなど）**…… カナ キーを押してカナモードにしてから、GRPH キーを押しながら、使いたい字のキーを押す。

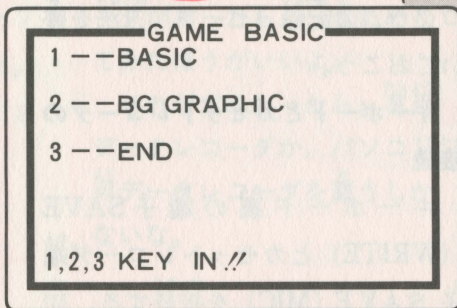
(1)　電源をONにする。画面に「カセットノバックアップスイッチヲ　OFFニ　シテクダサイ」と表示される。

(2)　バックアップスイッチをOFFにする。

■ BASICモードのとき

BASICモードのとき、それまで使っていたベーシックのプログラムを残すには、次の方法でバックアップスイッチONだ。

(1)　[S][Y][S][T][E][M][↵]とする。画面がGAME BASICモードとなる。

```
GAME BASIC
1 ーーBASIC
2 ーーBG GRAPHIC
3 ーーEND

1,2,3 KEY IN !!
```

キーボードの使い方

特殊キーの使い方

[STOP]キー、[RETURN]キー、[DEL]キーなど、ただ文字を入力するだけではなく、特殊なはたらきをするのが特殊キーだが、ここでは[RETURN]キーだけを説明しておく。他のキーはあとでね。[RETURN]キーの役目は、

(2) ファミリーベーシックのカセットのバックアップスイッチをONにする。

(3) 「3……END」を選択する。画面はスタート画面になる。

(4) ファミリーコンピュータの電源スイッチを切る。

＊メモリバックアップ使用の注意

電池の交換は、カセットを本体に差し込んでいるときに行なう。本体から抜いたまま交換すると、データは消えてしまう。

ベーシックモードを実行すると、各ボードの画面のデータは消える。

各ボードを実行すると、ベーシックプログラムのデータは消える。

●カセットレコーダにセーブ●

せっかく苦労して作ったプログラムは記録して残しておきたいものだ。ここでは、カセットレコーダへのプログラムの記録（セーブ）方法を書いておこう。

■キーボードとカセットレコーダの接続

キーボード裏の端子SAVE(WRITE)とカセットレコーダ端子SAVE(MIC)を接続する。同

キーボードの使い方

入力の合図だ。プログラムを入力するとき、各1行を入力し終わったらかならず押す。また、入力したプログラムを実行させるときも忘れずに押す。

カーソルキーの使い方

カーソルキーには▲▼◄►の4

RETURN キーは
省略形として
⏎と書くことが
あるから
おぼえておいてね

じようにLOAD(READ)とLOAD（EAR）を接続する。

セーブした
レコーダと
同じレコーダで
ロードするんだよ

■プログラムのセーブ

　メモリに記録されているプログラムを、カセットテープに記録して保存する方法だ。カセットテープの用意はできたかな？

(1)　ダイレクトモードで、
　　SAVE"ファイルネーム"
　と入力する。

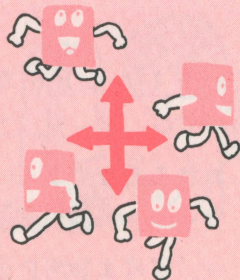
＊カセットレコーダの選択

　カセットレコーダなら、なんでもいいというわけではないよ。ファミコンベーシックって、けっこう気むずかし屋なのだ。

　レコーダの機種によっては、セーブ(保存)やロード（呼び出し）ができないものもあるから、手持ちのカセットレコーダがあったら、一度ためしてみたほうがいいよ。

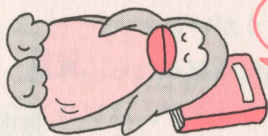
　それがだめだったら、専用データレコーダか、パソコン用データレコーダを買うしかないな。

キーボードの使い方

個のキーがある。画面に表示されたカーソル（■または▬）の位置は、押したカーソルキーの矢じるしの方向に移動する。なお、キーを押して入力した文字や記号は、画面のカーソルのある位置に表示される。

(2)　カセットレコーダの録音ボタンを押す。

(3)　録音状態になったら RETURN キーを押す。プログラムの記録中は、画面に、

WRITING　〝ファイルネーム〟

と表示されている。

＊ファイルネーム

ファイルネームはいわばそのプログラムの名まえだ。なんという名にするかは、自分で考えるんだな。16文字以内なら、どんな名まえでもいい。

```
SAVE　〝シンセイ〟
WRITING　シンセイ
```

(4)　セーブが終了するとOKと表示される。OKと出たら、レコーダを止める。

これでプログラムのセーブはいちおう終わった。だけど、ほん

キーボードの使い方

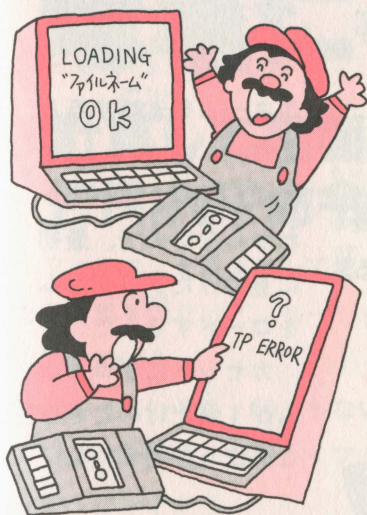
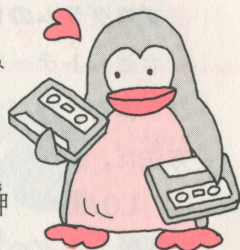
ファンクションキーの使い方

F1 から F8 までの8個のキーがファンクションキーだ。これらのキーには、よく使われる命令やことばが入っている。たとえば、スタートの選択画面では、「ハイ」が F1 キー、「イイエ」が F2 キーだったね。また、カ

とうに正確に記録されているかな？

正確に記録されているかどうか、たしかめてみよう。

(5)　カセットテープを巻き戻す。

(6)　LOAD? ⏎ とし、レコーダの再生ボタンを押す。

LOADING　〝ファイルネーム〟

OK

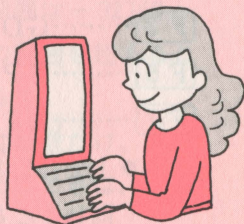
と表示されていたら、正しくセーブされたしるしだ。

？TP ERROR

と表示されたら、正確にセーブされていないしるしだから、音量を変えたり、テープを別なものと交換したりして、再度セーブの操作をしてみよう。

キーボードの使い方

リキュレータボードのときは、演算記号＋－×÷の代わりをするよ。

ベーシックを起動したとき、ファンクションキーに入っている命令は、次のようになっている。

● F1 ………LOAD(M)　　● F2 ………PRINT

■プログラムのロード

カセットテープにセーブしたプログラムを、呼び出して使用する操作方法だ。

(1) LOAD "ファイルネーム" ⏎

(2) レコーダの再生ボタンを押す。ロードの動作が始まる。ロード中は画面に、

LOADING "ファイルネーム"

と表示されている。

(3) ロードが終了するとOKと表示されるから、レコーダを止める。

＊ファイルネームの省略

LOAD ⏎ とファイルネームを省略すると、どのプログラムをLOADすればいいのかわからないので、最初に見つけたプログラムをロードするんだ。

カセットにプログラムが１個だけのときは、このほうが便利だね。

キーボードの使い方

- F3 ………GOTO
- F4 ………CHR$(
- F5 ………SPRITE
- F6 ………CONT(M)
- F7 ………LIST(M)
- F8 ………RUN(M)

※上で(M)とあるのは、⏎（RETURN キー）を押したと同じことだ。

ステップ　3

ファミコンを電卓にしちゃう

♥計算して画面に表示するのはPRINT命令だ

ファミコンを電卓にして使っちゃおう。カリキュレータボードなら計算専門だけど、ベーシックでも計算はできるんだ。

25＋25は？

「わかっているよ、50じゃないか」などと暗算でしてしまったらカワユクナイぞ。とにかくファミコンにやらせてみよう。

キーボードから、

PRINT　25＋25⏎

と入力してみよう。

⏎はRETURNキーと同じ

```
PRINT  25+25
 50
OK
■
```

＊ベーシック計算の注意

1．扱える数の範囲は－32768～32767まで。この範囲を超えるとヘンな数が出るよ。

2．扱える数字は整数だけ。割り算で小数点が出たときは、小数点以下切り捨てだ。

文字や数字を画面に表示だよ

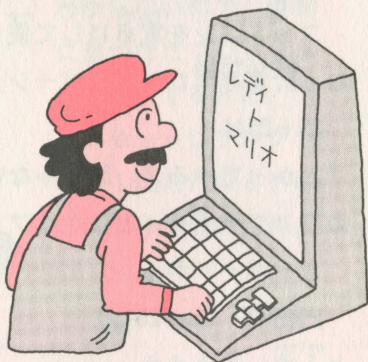
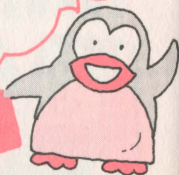
♥PRINTはPRINTでも" "で囲んだ文字を表示させる

キーボードがあってテレビがある。キーボードから入力した文字や数字を、テレビに表示させられなければ、なんのためのキーボードか、てことになるね。

テレビの画面に表示させる前に、画面をそうじして、きれいにしておこう。

●画面をきれいにそうじしてからPRINTだ

SHIFT キーと CLR／HOME キーを同時に押してみよう。

これが画面のそうじだ。それまで画面になにかが表示されていたとしても、みんな消えてしまって、

CLR／HOME キーだけを押せばカーソルを左上の位置に移動させる

これですっかりきれいになったわけだ。それから、またも出ましたPRINT命令だ。「BASIC」という文字を表示させてみよう。

PRINT “BASIC” ⏎


```
PRINT "BASIC"
BASIC
OK
■←カーソル
```

PRINT の次に入力した“ ”(ダブルクォーテーション)で囲まれた文字が、そっくりそのまま表示される。PRINT命令は、文字や数字を画面に表示させる命令なのだ。

前のステップ2でも、PRINT命令が出てきたね。いったいどう違うのだろう？

＊PRINTの省略形

PRINTは、ただの？で代行させることもできる。

PRINT “BASIC”

は、？を使って、？ “BASIC” でも同じなのだ。

PRINT 25＋25 ⏎ としたら、50とその答えが表示されたのだったね。

```
PRINT "25+25"
25+25
OK
■←カーソル
```

PRINT “25＋25” ⏎

と、“ ”で囲んで実行させてみよう。

“ ”で囲まないただの数式は、そのまま計算して、計算結

果が表示されたし、〝〟で囲んだ数式は、数式そのものが表示されるのだ。この違いは、もちろんわかったね。

●入力ミスはよくあること。正しく訂正すればいい

キーボードから入力していて「あ、まちがえちゃった！」なんてよくあること。気にしない気にしない。正しく訂正すればどうってことないんだから。

では、まちがえてみよう。直してみよう。

まちがいのある行を始めからすっかり打ち直してもいいよ

●1文字打ちまちがえの訂正

```
ABCDEFH
```

もちろん、ABCDEFときて、Gを打つつもりで、うっかりHを入力してしまったという場合だ。

カーソルキーで、カーソルをHのところに移動しよう。

```
ABCDEF[H]  ←カーソル
```

正しいGのキーを押そう。

```
ABCDEFG[ ]  ←カーソル
```

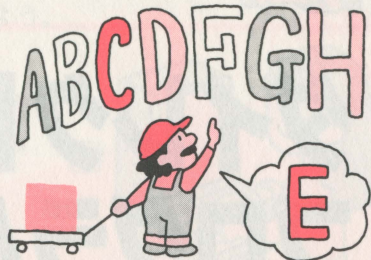
●1文字打ち忘れた追加訂正

```
ABCDFGH[ ]  ←カーソル
```

あ、DとFのあいだにEを入れ忘れた、Eを追加しなければ、という場合だ。

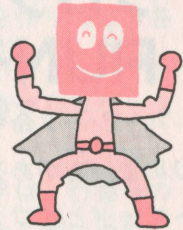
カーソルキーで、カーソルをFの文字の位置（Dの後ろ）に移動しよう。

カーソル
ABCDFGH

INS キーを押そう。

カーソル
ABCD　FGH

カーソルの位置にあるFの字が右にずれて、カーソルの位置に1字分空白ができる。そこで、追加するEのキーを押そう。

カーソル
ABCDEFGH

● 1文字余分の字を入力してしまった削除訂正

ABCDDEFGHI

Dは2個いらないから、Dを1個削除してしまおう。カーソルキーで、カーソルをEの位置（削除したいDの後ろ）に移動させよう。

カーソル
ABCDDEFGH

DEL キーを押そう。

カーソル
ABCDEFGH

カーソルの左の文字(D)が消え、E以下が左に1つずつ上がって、ABCDEFGHと、ちゃんとした並びになったはずだ。

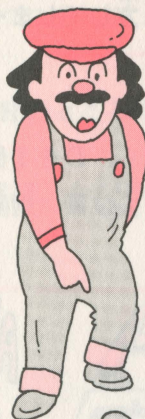
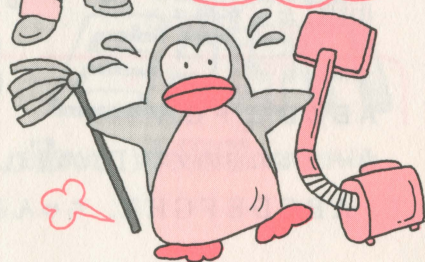
ステップ　5

ダイレクトモードとプログラムモード

♥プログラムの実行はRUNそしてRETURNだ

ベーシック命令には、2通りの使い方がある。ダイレクトモードとプログラムモードだ。

ダイレクトモードは、命令文を入力して、↵とすると同時に実行するモード。

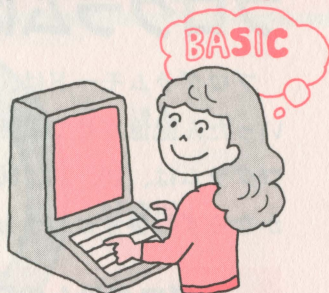
前で勉強したPRINT〝BASIC″という命令文、おぼえているかな？ RETURN キーを押すと同時に実行して、画面にBASICと表示したね。これがダイレクトモードなのだ。

●BASICと表示させる3行のプログラム

次のプログラムを入力してみよう。

```
10  CLS ⏎
20  LOCATE  13, 12 ⏎
30  PRINT  "BASIC" ⏎
```

RETURN キーを押すたびに、命令文が画面に表示されるけど、ただそれだけ。PRINT "BASIC"とあるのに、BASICと表示されないではないか。というわけで、こんどはRUNと入力して⏎だ。

画面が1度消えてから、画面のほぼ中央にBASICと表示されたね。これがプログラムモードの実行なのだよ。

BASIC

＊プログラムと行番号

プログラムを作りあげる各命令文は、1行ごとに文頭に行番号をつけなければならないんだ。

コンピュータは、行番号の小さな命令文から順に実行していくから、はじめは小さな行番号から始めて、だんだん大きな行番号にしていく。

行番号は、命令の各文ごとに、たとえば、1、2、3……でも、100、200、300……でもいいけど、とりあえず最初の行を10として、20、30、40……と10きざみに付けていったほうがいいよ。なに、理由はそのうちにすぐわかるさ。

プログラムは1行ごとに行番号

プログラムモードは、ベーシックの命令文をプログラムとしていったん記憶させておき、あとでまとめて実行する。それからプログラムは、10、20、30……というように、1行の命令文ごとにかならず行番号というものが付いていることに注意してほしい。

■プログラムのRUN、LIST、それから……

■ RUN 命令

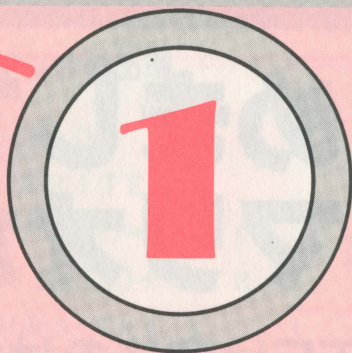
RUN と入力して↵とすれば、行番号の小さい順に1行ずつプログラムの命令を実行していく。「プログラムよ。動け！」という命令だ。

■ LIST 命令

メモリに記憶してあるプログラムを画面に表示しなさいという命令。入力したプログラムにまちがいはなかったかな、などと、気になるときに、LIST ↵とすれば、行番号順にプログラムを画面に表示させてくれるよ。

■ NEW 命令

それまでメモリに記憶してあるプログラムをすっかり消してしまう命令。これから新しいプログラムを入力しようというとき、古いプログラムを消してきれいにするために使われるよ。ただし、まだ必要なプログラムを NEW 命令で消してしまって、あわてたりするなよ。

パート1

マリオ、画面に出てこい

ステップ　1

はじめまして ぼくマリオだよ

♠SPRITE(スプライト)は魔法(まほう)の杖(つえ)だ キャラを自由(じゆう)に出現(しゅつげん)させる

アニメキャラクタマップAを見(み)てごらん。いるぞ、いるぞ。なんともカワユクて愉快(ゆかい)なアニメキャラクタが、ゾロゾロいるではないか。マリオにレディ、ペンペン、ニタニタ……。

こんなに楽(たの)しいアニメキャラクタたちを画面(がめん)に呼(よ)び出(だ)して、歩(ある)かせたり、遊(あそ)ばせたり、ケンカさせたりしたら、ファミコンベーシックが、いっぺんで好(す)きになってしまいそう。

そうだ、やるっきゃないのだ。

文句なし。次のプログラムでレッツゴー！だ。

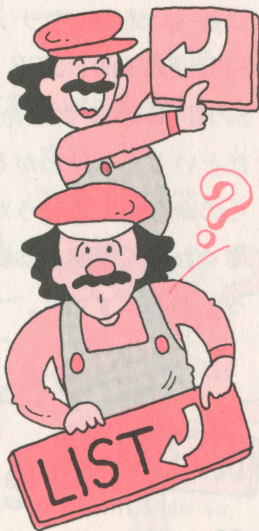
●画面の中央にマリオの出現

```
10  SPRITE  ON
20  DEF  SPRITE  0,
  (0, 1, 0, 0, 0)=CHR$
  (0)+CHR$+(1)+CHR
  $(2)+CHR$(3)
30  CLS
40  SPRITE  0, 100, 100
```

プログラムは
1行入力したら
忘れずに⏎だよ

入力したリストに
自信がなかったら
LIST⏎で
画面に表示させて
たしかめよう

プログラムにまちがいはないかな？　ゼッタイまちがいないと自信マンマンなら、そのままRUNとして⏎だ。

どうなったかな？　画面のほぼ中央に、アニメキャラクタテーブルAのマリオが表示されたね。

いまはマリオが出るだけでいい

短いけど、むずかしそうなプログラムだって？　そう、そうなんだよ。たった4行のプログラムだけれど、この中にたくさんの問題が詰め込まれているんだ。

でもいまは、あまりプログラムのセンサクはよそう。ここではただ、このプログラムを実行すれば、マリオが出てきたゾと、おもしろがっているだけでいいよ。

このプログラムでは、SPRITE ON、DEF SPRITE、SPRITEの3つの命令が中心だということはわかるね。これらの命令をどう使うかは、あとでくわしく説明する機会が、きっとあるはずだよ。

＊SPRITE関係命令の意味

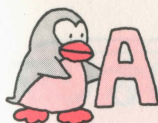
SPRITE ON

スプライト画面にキャラクタを表示させるときには、かならず実行しなければならない決まり文句だ。

DEF SPRITE

キャラクタテーブルAにあるどのアニメキャラクタを、どんな状態で表示させるかを決める。スプライトの定義というんだよ。

SPRITE

DEF SPRITEで定義したキャラクタを表示させる位置を決めて、実際に表示させる命令だ。

ステップ　2

文字に重なってこんにちはマリオ

♠PRINTはバックグラウンド SPRITEはスプライト面

前にPRINT文で、文字を画面に表示させたね。思い出してみよう。

PRINT　"BASIC" ↵

と、こうすれば、画面にBASICという文字が表示されたのを、おぼえているだろう。

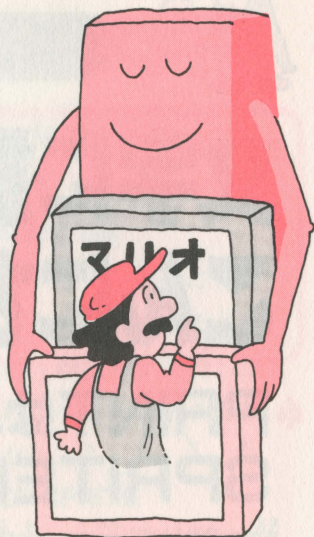
そしてさっきは、マリオのアニメキャラクタが画面に出てきたね。「ステップ1」のプログラムの40行に、

文字などが表示される画面とマリオやキャラが表示される画面はちがう画面だってホント？

40　SPRITE　0, 100, 100

とあったはずだ。

文字(もじ)もキャラクタも、同(おな)じテレビの画面(がめん)に表示(ひょうじ)されていたけど、ほんとうのことをいうと、文字(もじ)が表示(ひょうじ)されていた画面(がめん)はバックグラウンド画面(がめん)、マリオが表示(ひょうじ)されていた画面(がめん)はスプライト画面(がめん)といって、区別(くべつ)されているんだ。

同(おな)じテレビの画面(がめん)なのに、どうして2つの画面(がめん)があるの？っていいたいのだね。まあ、その証拠(しょうこ)をお目(め)にかけよう。

●文字(もじ)の上(うえ)に重(かさ)なったマリオ

```
10　SPRITE　ON
20　DEF　SPRITE　0,
    (0, 1, 0, 0, 0)=C
    HR$(0)+CHR$(1)
    +CHR$(2)+CHR$
    (3)
30　CLS
35　LOCATE　9, 11:P
    RINT　"AAAAA"
40　SPRITE　0,100,105
```

マリオのかげにAの字がかくれた！

入力したら例によって，RUN [↵] だよ。

AAAAAという文字の上にマリオが重なって、しかも、重なった部分は消えているけど、マリオは全身すっかり表示されているね。

これは、AAAAAが表示されているバックグラウンド面の上に、マリオが表示されているスプライト面が重なっているのだよ。

これで、文字などを表示するバックグラウンド面と、アニメキャラクタを表示するスプライト面の2枚の画面があることが、わかったね。

バックグラウンド面

AAAAA

スプライト面

前面にスプライト面を出す

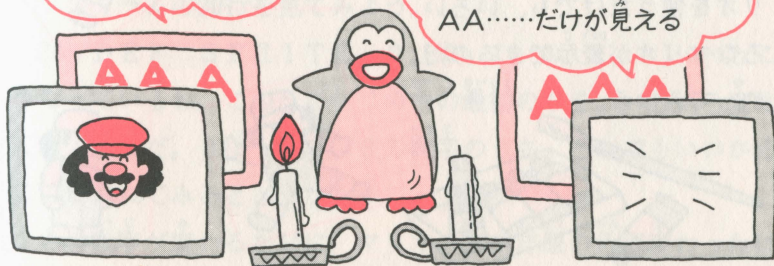
■SPRITE ONはマリオの舞台

スプライト画面とバックグラウンド画面は、わかったよね。バックグラウンド画面は文字を表示させる画面、スプライト画面はアニメキャラクタが表示される画面だ。そして、SPRITE ONは、スプライト画面に描かれたアニメキャラクタが見えるように、スプライト画面にあかりを入れる命令だ。

（バックグラウンド面）　（スプライト面）

AAAAAAAAA

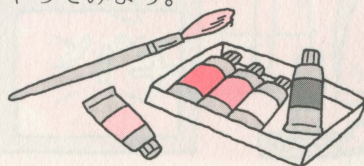
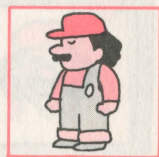
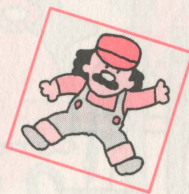
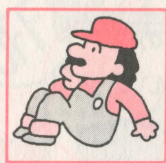
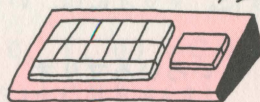
SPRITE ON　SPRITE OFF

AAAAAAAAAA

ステップ 3

変わるよ変わるいろいろマリオ

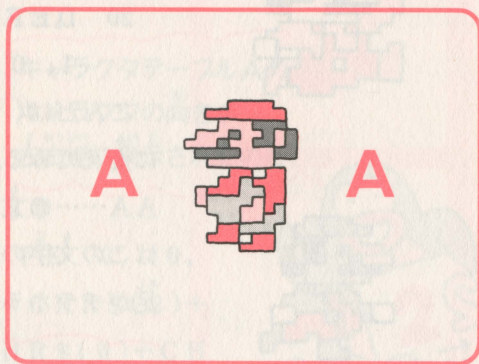
♠DEF SPRITE文の数字を変えれば表示も変わる

キャラクタテーブルAには、いろいろなマリオがいるから、そのうちのどのマリオを呼び出すかで、マリオのキャラはいろいろ表示させることができる。これはあたりまえだ。

でも、キャラクタテーブルAにあるどれか1個だけのマリオを使うだけでも、いろいろなマリオが表示できるのだ。やってみよう。

とりあえず、前に出たプログラムをちょっとだけ変えたものを、もう一度入力して、と。

```
10 SPRITE ON
20 DEF SPRITE 0,(0, 1, 0, 0, 0)=CHR$(0)+CHR$(1)+CHR$(2)+CHR$(3)
30 CLS
35 LOCATE 9, 11:PRINT "AAAAAAAA"
40 SPRITE 0, 100, 105
```

と、こうだったね。

RUN ⏎ だ。

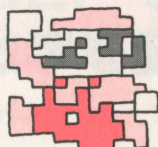
画面のほぼ中央に、左向きのマリオくん（WALK1）がいて、そのうしろのAの字が、ちょっと見えたりしているね。

■色が変わって4通りのマリオ

プログラムの20行を見てみよう。

20 DEF SPRITE 0,(0, 1, 0, 0, 0)……とこうなっているね。このうち(　)の中の最初の0を、いろいろ変えてみて実験だ。0、1、2、3の数字のうち、どれでもいいから、ここに入れてみるとどうかな？

この数字が変わるたびに、マリオの顔や洋服の色が変わったは

ずだ。マリオには、いろいろなポーズや衣裳があるのだ。

いろいろマリオ

いちばん上が
配色番号0のマリオ
あとは1、2、3の順だ
配色が変われば
感じも変わるものだなあ！

■文字の後ろにカクレンボのマリオ

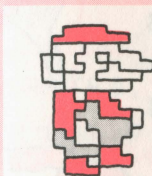
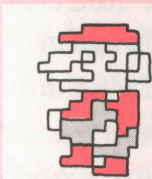
また、20行だ。

20 DEF SPRITE 0,(0, 1, 0, 0, 0)……

こんどは( )の中の3番目の0を1に変えてみる。どうなったかな？ AAA……の文字が表面に出て、マリオはこの文字のかげにカクレンボ。頭や足がチラホラ見えるけどね。

バックグラウンド面が前に出て
スプライト面が後ろにかくれたのだ

■くるりと〝マワレーミギ！〟のマリオ

```
20 DEF SPRITE 0,(0,1,0,1,0)=CHR
   $(1)+CHR$(0)+CHR$(3)+CHR$(2)
```

左右逆のマリオ

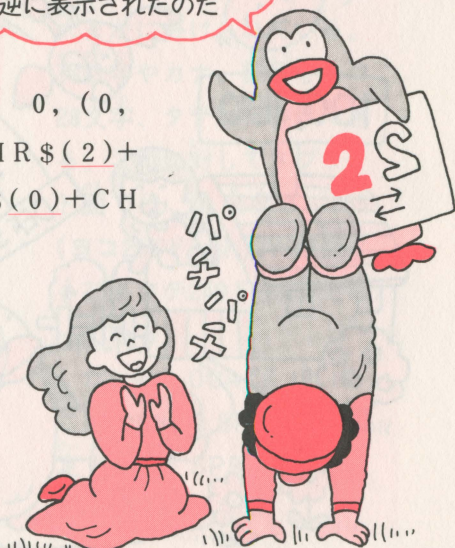
（　）の中の４番目を０から１に変えるだけじゃないよ。＝の次のＣＨＲ＄（ナンタラカンタラ……）とつづく（　）の中の順序に注目しよう。

前は（０）、（１）、（２）、（３）の順序だったはずだけど、これは（０）と（１）の順序も入れ替えて、結局（１）、（０）、（３）、（２）とするんだ。

キャラクタテーブルＡのキャラクタの向きが左右逆に表示されたのだ

■逆立ちだってできるマリオ

```
20 DEF SPRITE 0,(0,
   1,0,0,1)=CHR$(2)+
   CHR$(3)+CHR$(0)+CH
   R$(1)
```

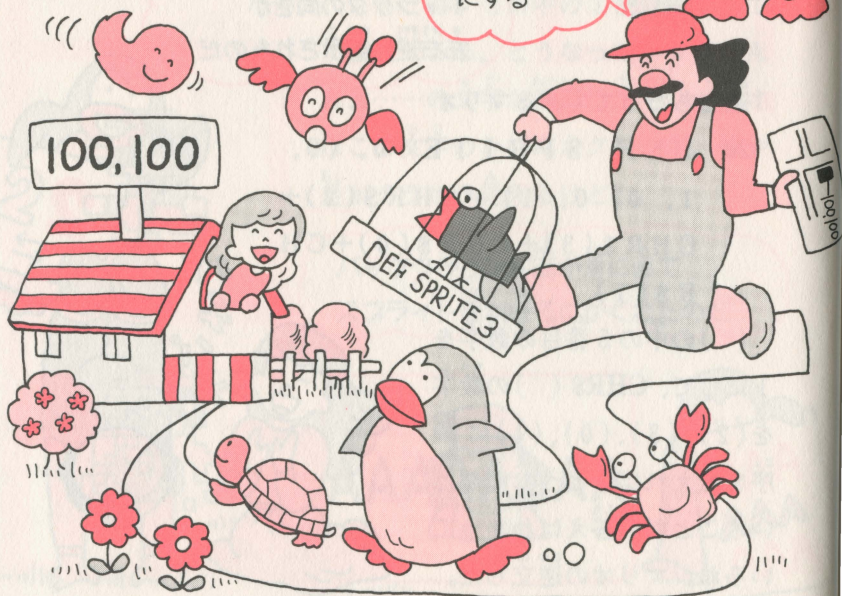
（　）の中の５番目の数字を１にして、CHR$（　）の数字を（２）、（３）、（０）、（１）に順序を変えれば、どうなるかということだ。答えはわかっているね。マリオの逆立ちだ。

■どこに表示させるかを決めるSPRITE

SPRITE 0，100，100

0はスプライトの番号

100，100というのは
画面の場所の
ことなんだよ

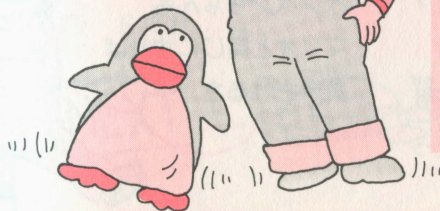
SPRITE　0というこの0は、スプライト番号0のスプライトという意味だよ。そしてこの0は、スプライトを定義するDEF SPRITE 0……とあった0と一致しなければならないのだ。

DEF
SPRITE　3
なら
SPRITE　3
にする

SPRITE 0, 100, 100 の100, 100は、スプライト面のX座標(ヨコ) 100、Y座標(タテ) 100の位置ということ。

SPRITE 0, 100, 100 をまとめてみよう。

DEF SPRITE で決めたキャラクタをX座標100、Y座標100の位置に表示せよということになる。

SPRITE 0，100，100
ヨコ100，タテ100の位置

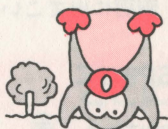
ぼくたちは
ここに
表示される
んだよ

＊LOCATE と SPRITE

LOCATE も SPRITE も、画面の中に表示させる位置を決める命令だけど、位置を指定する数値の単位は違うから、注意してくれ。

LOCATE を使うバックグラウンド面と、SPRITE を使うスプライト面では、位置を示す単位が違うからなのだ。

バックグラウンド面は、X座標（ヨコ）が0～27、Y座標（タテ）が0～23と大まかに分けられている。英数字やカナ、記号をヨコ28文字、タテ24行表示できる。これに対してスプライト面では、X座標0～255(ヨコ256ドット)、Y座標0～239(タテ240ドット)となっている。

だから、LOCATE 10, 10 と同じ位置をSPRITEで指定すると、SPRITE 0, 96, 104 ぐらいになる。

ステップ 4

動いて逆立ち歩くよマリオ！

♠レディちゃんゴメンナサイ マリオをどんどん動かせ！

ここまでは、キャラクタテーブルAのアニメキャラクタを表示させる方法を、マリオくんに代表してもらって説明してきた。

SPRITE ONして、DEF SPRITEして、SPRITEするんだったね。

でも、ただ画面の中央にアニメキャラを表示させるだけなんて、なんだかとってもツマラナイみたい。もっとなんとかならないの？って声も聞こえてくるようだ。

まあ、待て待て、アルプスを登るのだって、下から一歩一歩登っ

て行かなくては、頂上にまでたどり着けないんだぜ。

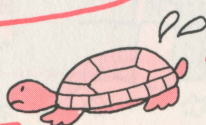
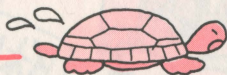
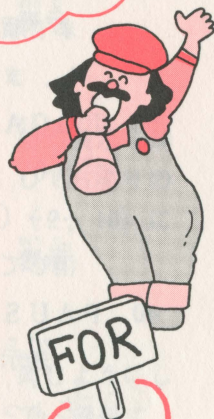
よし、ここではあまりクドクドと説明するのはよそう。マリオを画面に出して、いろいろなことをやらせるプログラムを、どんどん出していくから、どんどん入力してRUN ⏎して、ナットクしてくれ！

レッツゴー、マリオ・バラエティだ。

●キャラクタをごく単純に動かす

最初はアニメキャラ抜きの練習プログラムだ。画面でAの文字を左から右に動かしてみる。アニメキャラは出てこないけれど、アニメキャラを動かすのも、このプログラムが基本なのだとあきらめてついてこい。

FOR～NEXTはくり返しのルーチンだ

```
10  CLS
20  FOR  I＝0  TO  27
30  LOCATE  I，10：PRINT "A"
40  PAUSE  10
50  LOCATE  I，10：PRINT " "
60  NEXT
70  END
```

♠動いているように見せるのは書いて消して次の位置に書く

```
20  FOR  I=0  TO  27
 ∫
60  NEXT
```

20行のFORと60行のNEXTにはさまれた30行〜50行を、0から27すなわち28回（I=0　TO 27）くり返せということだ。

なにをくり返すの？

```
30  LOCATE  I, 10:PRINT "A"
```

座標位置（0, 10)、(1, 10)、(2, 10)……(27, 10)まで順に〝A〟という字を表示し、また、

```
50  LOCATE  I, 10:PRINT " "
```

(0, 10)、(1, 10)、(2, 10)……(27, 10)の位置の表示を消していけ。ただし、

```
40  PAUSE  10
```

表示させて，表示を消すまで、ちょっと時間をとれ、というわけだ。

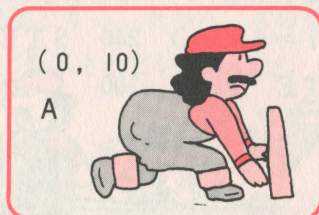
PRINT〝 〟はなにもないもの(空白)を表示せよだから表示を消すことだ

PAUSEは時間かせぎの命令だ
数値が小さくなれば休み時間は短くなる

ここでは、表示させることと、その表示が動くように見せることだが、そのテクニックは簡単だ。ヨコ（X座標）に動く場合なら、(1)ある位置に表示させる──→(2)表示を消して隣りの位置に表示させる……という動作をくり返せばいいのだ。

■**動いたように見えるコツ**

ヨーイ！
スタート位置につくマリオ

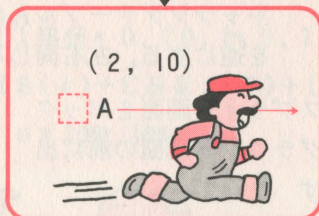
(0, 10)
A

(0, 10) の位置に表示

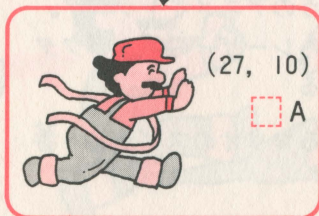
ドン！
動きだすマリオ

(1, 10)
A

前の表示を消して右隣りの(1, 10) に表示

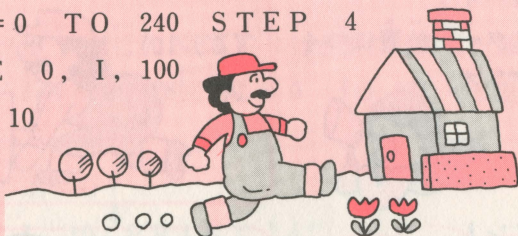
走りつづけるマリオ

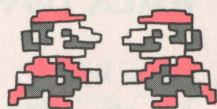
(2, 10)
A

前の表示を消して右隣りの(2, 10) に表示

ゴールのテープを切るマリオ

同じようにして、最後に、(27, 10) に表示

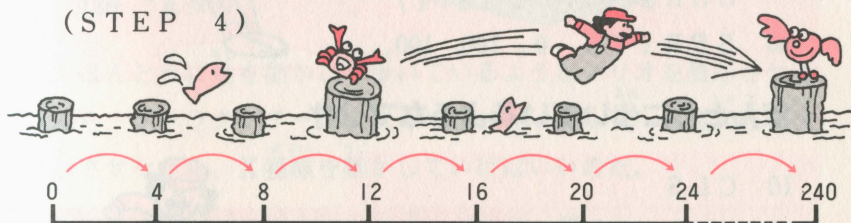
●マリオを左から右に動かす

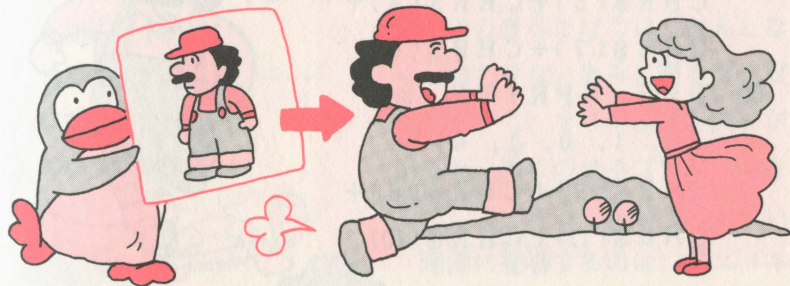
```
10 CLS
20 SPRITE ON
30 DEF SPRITE 0,(0,1,0,1,0)=CHR
$(1)+CHR$(0)+CHR$(3)+CHR$(2)
40 FOR I=0 TO 240 STEP 4
50 SPRITE 0, I, 100
60 PAUSE 10
70 NEXT
80 END
```

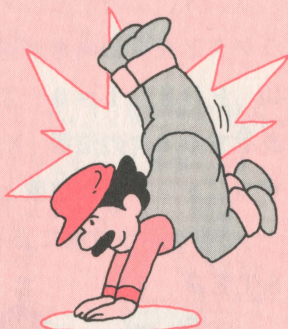
```
30 DEF SPRITE 0,
(0, 1, 0, 1, 0) =
```

- 5番目の0：キャラクテーブルAのマリオと上下の向きは同じ
- 4番目の1：キャラクテーブルAのマリオと左右の向きを逆にする。左右同じなら0
- 3番目の0：スプライト画面をバックグラウンド画面の前に出す
- 2番目の1：ふつうのアニメキャラクタ。0ならレーザーのキャラクタになる
- 1番目の0：表示させるマリオのスプライト番号は0

DEF SPRITEのCHR$（数字）の順序は、１、０、３、２だから、テーブルのアニメキャラクタと左右の向きが逆のマリオだ。

40　FOR　I＝0　TO　240　STEP　4

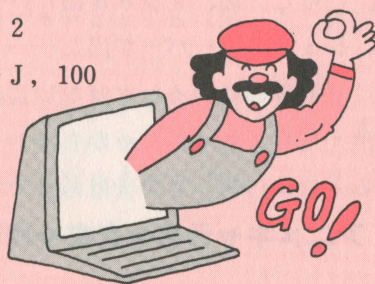
表示するヨコの位置（X座標）は、０～240までだが、ただし４つずつジャンプした位置とする。だから、０の次は４、８、12、16……240という位置に表示されていくわけだ。

（STEP　4）

WALK２のマリオを右向きにする

```
10　CLS
20　SPRITE　ON
30　DEF　SPRITE　0,(0,1,0,1,0)=CHR
    $(5)+CHR$(4)+CHR$(7)+CHR$(6)
40　SPRITE　0,100,100
```

●WALK 1のマリオに逆立ちさせる

```
10  CLS
20  SPRITE ON
30  DEF SPRITE 0,
     (0, 1, 0, 0, 1) =
    CHR$(2)+CHR$(3)+
    CHR$(0)+CHR$(1)
40  SPRITE 0, 100, 100
```

●ほんとうに歩いているようなマリオ

```
10  CLS
20  SPRITE ON
30  DEF SPRITE 0,
     (0, 1, 0, 1, 0) =
    CHR$(1)+CHR$(0)+
    CHR$(3)+CHR$(2)
40  DEF SPRITE 1,
     (0, 1, 0, 1, 0) =
    CHR$(5)+CHR$(4)+
    CHR$(7)+CHR$(6)
50  DEF SPRITE 2,
     (0, 1, 0, 1, 0) =
    CHR$(9)+CHR$(8)+
    CHR$(11)+CHR$(10)
```

```
60 FOR I=0 TO 240 STEP 6
70 FOR J=0 TO 2
80 SPRITE J, I+J, 100
90 PAUSE 5
100 SPRITE J
110 NEXT:NEXT
120 END
```

ほんとうに足を動かして歩いているようなマリオを表示させるなら、キャラクタテーブルAのマリオWALK 1～3を交互に表示させながら、X座標を動かしていけばいいのだ。

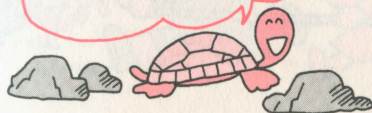
おもしろいかおもしろくないか　遊んでプログラム

アニメキャラ16個が総登場

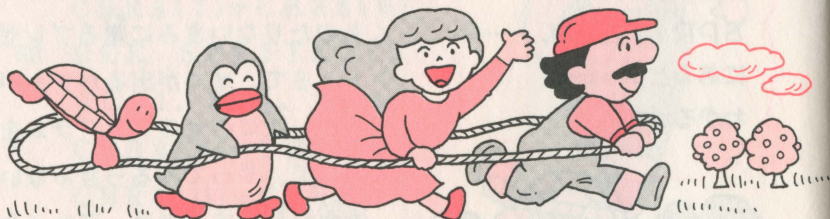
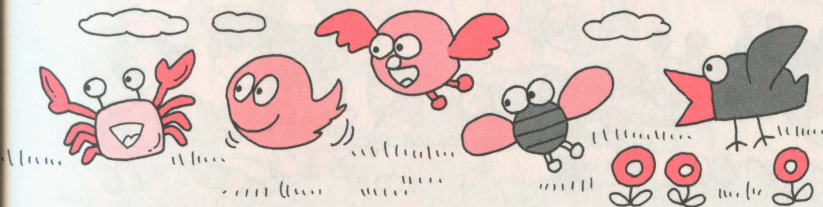
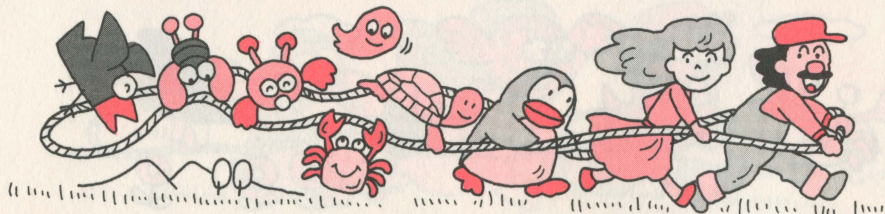
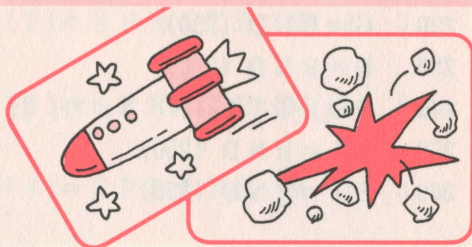
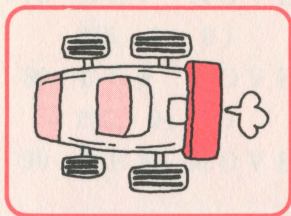
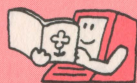
アニメキャラ
16個表示のポイントは
プログラムの40行
FOR M=0 TO 15
と50行以下の
SPRITE (M, ………
にあるといったら
わかるかな？

動いたり逆立ちしたり歩いたりのマリオを表示させるプログラムの応用編だ。

マリオ、マリオと、いつもマリオだけの表示だけでは、なんとなくものたりないきみに贈るプレゼント。さて、オニが出るかジャが出るか。次にのせるプログラムを入力して、やってみるっきゃない。

画面に登場するのは、アニメキ

ャラクタテーブルにある16個のキャラクタ全員だ。だけど、「あ、マリオが出た。ペンペンが出た。シェルクリーパーが出た——」などと、ゆっくり観賞しているヒマはないよ。表示されたり消えたり、ちゃかちゃかちゃかちゃか目がまわる。なにしろ、この世はめまぐるしいのだ。

アニメキャラちゃかちゃか表示。目がまわる

```
10 CLS
20 CGSET 1, 0
30 SPRITE ON
40 FOR M=0 TO 15
50 DEF MOVE(0)=SPRITE(M, 1, 3, 255,
   0, 0)
60 DEF MOVE(1)=SPRITE (M, 1, 3,
   255, 0, 0)
70 DEF MOVE(2)=SPRITE (M, 1, 3,
   255, 0, 0)
80 DEF MOVE(3)=SPRITE (M, 1, 3,
   255, 0, 0)
90 DEF MOVE(4)=SPRITE (M, 1, 3,
```

```
    255, 0, 0)
100 DEF MOVE(5)=SPRITE (M, 1, 3,
    255, 0, 0)
110 DEF MOVE(6)=SPRITE (M, 1, 3,
    255, 0, 0)
120 DEF MOVE(7)=SPRITE (M, 1, 3,
    255, 0, 0)
130 A=RND (250)
140 B=RND (230)
150 POSITION 0, A, B
160 C=RND (250)
170 D=RND (230)
180 POSITION 1, C, D
190 E=RND (250)
200 F=RND (230)
210 POSITION 2, E, F
220 G=RND (250)
230 H=RND (230)
240 POSITION 3, G, H
250 A1=RND (250)
260 B1=RND (230)
```

```
270  POSITION 4, A1, B1
280  A2=RND (250)
290  B2=RND (230)
300  POSITION 5, A2, B2
310  C2=RND (250)
320  D2=RND (230)
330  POSITION 6, C2, D2
340  E2=RND (250)
350  D2=RND (230)
360  POSITION 7, E2, D2
370  MOVE 0, 1, 2, 3, 4, 5, 6, 7
380  PAUSE 20
390  FOR I=1 TO 2000:NEXT
400  ERA 0, 1, 2, 3, 4, 5, 6, 7
410  NEXT
420  END
```

パート2

アニメキャラといっしょ

ステップ 1

マリオ歩いて MOVE、MOVE

◆DEF MOVE、POSITION、MOVEは連続ワザだ

キャラクタテーブルAのアニメキャラクタは、マリオだけじゃないよ。レディもスターシップもアキレスもペンペンもいる。まだまだたくさんいるから、どのアニメキャラクタと遊ぼうと、きみの自由だ。

SPRITE命令で、アニメキャラを表示さ

```
10 CLS
20 SPRITE ON
30 DEF MOVE(0)=SPRITE (0, 3, 1,120,
   0, 0)
40 POSITION 0, 0, 100
50 MOVE 0
60 END
```

せる方法をおぼえたら、こんどは、MOVE命令でマリオを動かしてみよう。

ここに新しく登場したのが、

DEF　MOVE、POSITION、

MOVEという命令だ。

DEF　MOVE

どんなアニメキャラを、どんなふうに動かすかを決める。

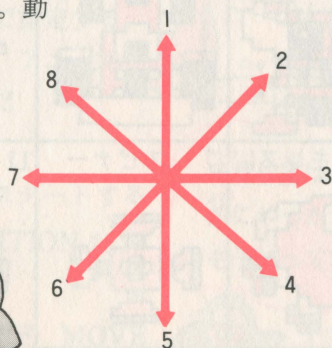
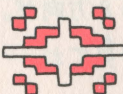
POSITION

動かすアニメキャラのスタートする位置はどこかだ。

MOVE

ヨーイ、ドン、スタート、動け！という命令だ。

■移動方向の番号

アニメキャラの移動方向は番号で示すんだよ。上下左右と右上、右下、左上、左下の8方向だ。動く方向の番号は図を見てね。

アニメキャラは背番号付きだ

キャラクタテーブルAのアニメキャラは16個のうち、どれでも表示させせることができるよ。

ひとりひとり、一匹一匹、一個一個のアニメキャラには、それぞれ0～15までの番号が決められているから、おぼえておこう。

■アニメキャラクタの番号

0　マリオ	4　ペンペン	8　スターキラー	12　レーザー
1　レディ	5　ファイアーボール	9　スターシップ	13　シェルクリーパー
2　ファイターフライ	6　車	10　爆発	14　サイドステッパー
3　アキレス	7　スピナー	11　ニタニタ	15　ニットピッカー

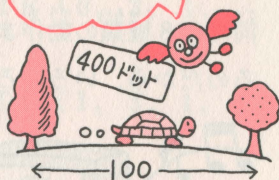
■DEF　MOVE、POSITION、MOVE

■DEF　MOVE命令

DEF MOVE (n) =
SPRITE (A, B, C, D, E, F)

- F：0〜3までの数字。4通りの配色が選べる
- E：0か1。0はキャラが前面に表示。1はバックグラウンドが前面に出る
- D：全移動距離。1〜255で指定。指定数の4倍のドットが移動距離
- C：移動するスピード。1〜255で指定。60ドット動くのにかかる秒数だ
- B：移動方向。0〜8の数字で指定する
- A：0〜15まで。アニメキャラの番号だ

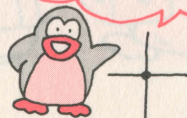
nは
0〜7までだ。
キャラの
動きに
番号を
付けるんだよ

移動距離100なら
100×4＝400で
400ドット移動だ

■POSITION命令

表示させたアニメキャラのスタートする位置を決めるのが、このPOSITIONだ。

POSITION　n, X, Y

nは動作番号だから、DEF　MOVE

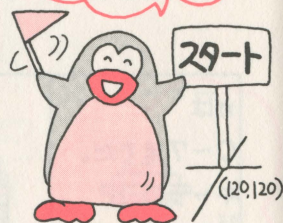
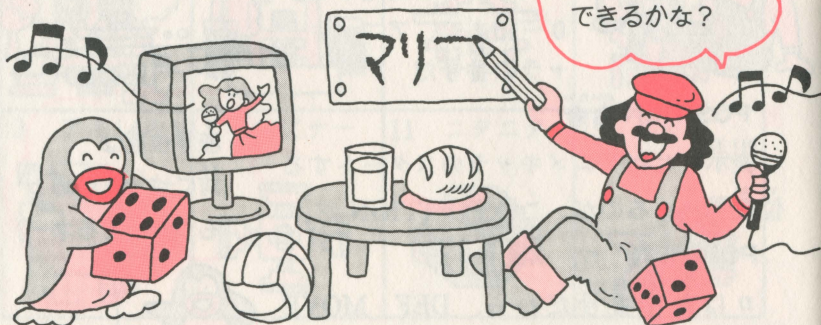
座標位置は
前にやったから
もう知ってるネ

0としたあとなら、ここも0で指定する。

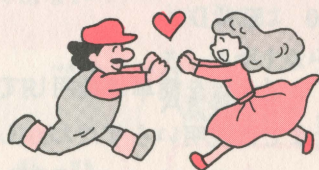
X，Yはスタート位置の座標だ。Xはヨコの座標、Yはタテの座標だから、Xは0～240，Yは5～220の範囲で指定する。

■MOVE命令

ホントに動けよ、オイ、という命令だ。MOVE　0なら、0という動作を始めよということだから、DEF　MOVE（0）としてあったら、これと同じ0で指定するんだ。0～7までの数字で、8種類の動作を同時にさせることができる。

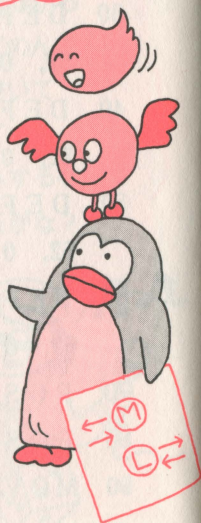
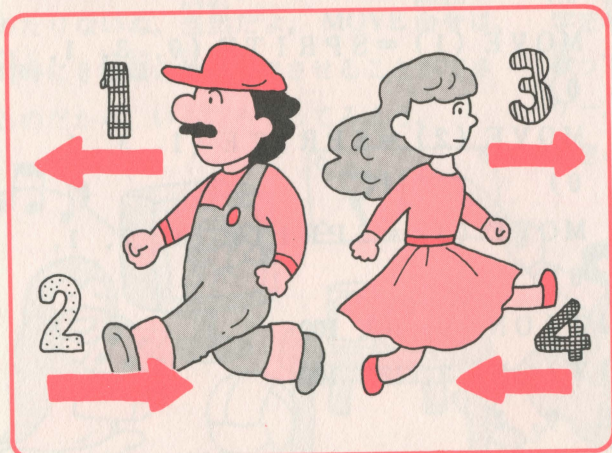
8種類の動作が同時にできる

アニメキャラクタを登場させて動かしても、1個のキャラをただ動かすだけではものたりないね。当然だよ。MOVE命令は、同時に8種類の動作までさせることができるのだから、いろいろやってみよう。

●別れたあとでまたデートするマリオとレディ

マリオとレディを同じ画面に表示させて、同時に動かしてみよう。RUN させると、画面の中央に立ったマリオとレディは、なにがあったのかプンプン。バイバイといったんは別れたのだが、やっぱりいっしょにいないとものたりなくて、またふたりでデートというしだいだ。

```
10 CLS
20 SPRITE ON
30 DEF MOVE(0)=SPRITE(0,7,1,
   52,0,0)
40 DEF MOVE(1)=SPRITE(0,3,1,
   52,0,0)
50 DEF MOVE(2)=SPRITE(1,3,1,
   52,0,0)
60 DEF MOVE(3)=SPRITE(1,7,1,
   52,0,0)
70 POSITION 0,104,100
80 POSITION 2,120,100
90 MOVE 0,2
```

```
100  IF MOVE(0)=-1 OR MOVE(2)
     =-1 THEN 100
110  POSITION 1, XPOS(0), YPOS(0)
120  POSITION 3, XPOS(2), YPOS(2)
130  ERA 0, 2
140  MOVE 1, 3
150  END
```

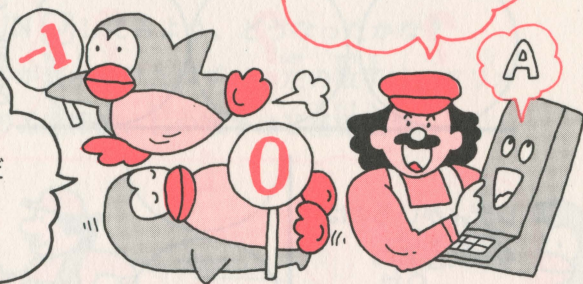
プログラムを簡単に説明しておこう。90行までは説明いらずだね。もう、知っていることだもの。ただ、DEF　MOVE文が4つ続いて、DEF　MOVE(0)~(3)となっているところが変わっている。これは、画面のアニメキャラの動作が4種類あるということだ。

マリオが左に行って
右に行き
レディが右に行って
左に行くから
合計4つの動作だ

◆もし〜ならという条件設定のIF〜THEN命令

100行のIF〜THEN命令は、はじめて出てくる命令だね。もし（IF）ナニナニなら、そのときは（THEN）こうしなさい、ということだ。そんなにむずかしくはないはずだ。

```
IF MOVE(0)=-1 OR MOVE(2)=-1 THEN 100
```

というわけだが、これは「もし、0番の動作または2番の動作が行なわれていたら、（−1）、100行に行け」という命令だ。

IF A=1
THEN
PRINT "A"
というのは
もしAが1なら
Aを表示せよ

ということだ

＊ ERA 命令

ERA　n1，　n2，…と使う。ある動作番号（n1，　n2……）の動作をするアニメキャラを画面から消す命令だ。

ERA　0，2なら0番の動作をするキャラと、2番の動作をするキャラを消せということだ。

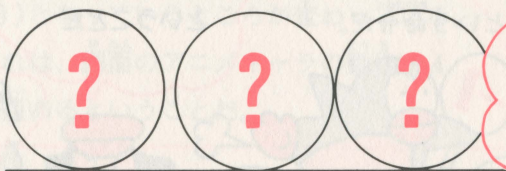
110行はちょっとややこしい。

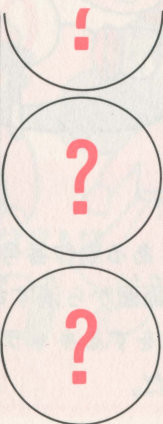
```
110 POSITION 1, XPOS
    (0), YPOS (0)
```

あとでくわしく説明するけど、ここでは、動作番号1が動きだす位置（POSITION 1）は、動作番号0番のキャラクタが表示されている位置（XPOS（0），YPOS（0））とする、とだけいっておこう。

XPOS（　）
YPOS（　）は
動きの方向転換によく使われるよ

■POS(　)は動きの座標を調べる

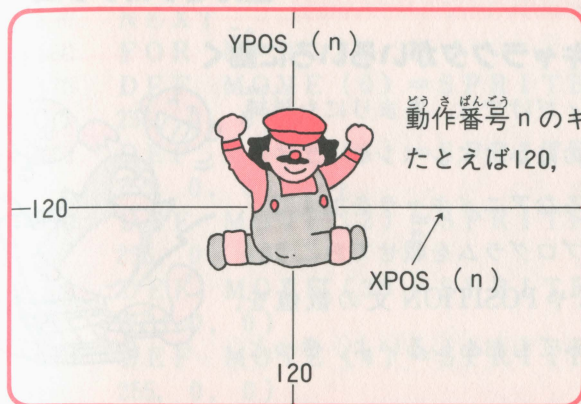
XPOS、YPOSは、いまアニメキャラクタが、どこに表示されているかを調べるものだ。

マリオがいるのはヨコの座標のどこだろう？

レディがいるのはタテの座標のどこだろう？

■ XPOS（n）

動作番号nの動きをするアニメキャラが、X座標のどこにいるかを調べる。

■ YPOS（n）

動作番号nの動きをするア

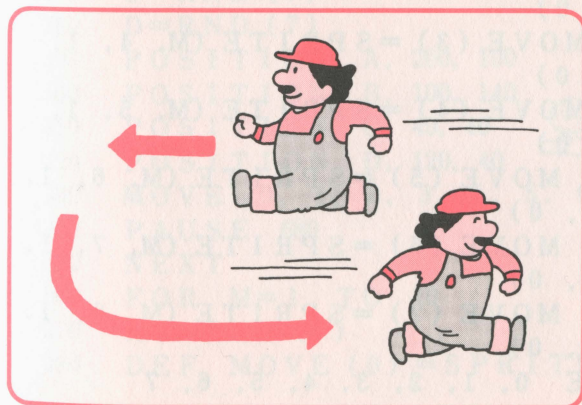
ニメキャラが、Y座標のどこにいるかを調べる。

ためしにプログラム追加

前のプログラムに、

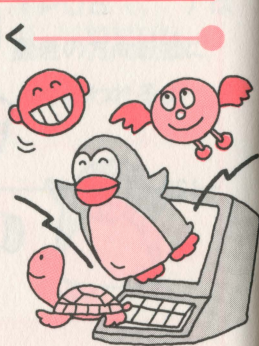
```
105  PRINT  XPOS (0),YPOS (0)
```

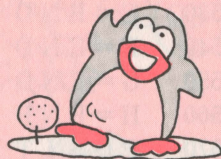
の1行を加えて、RUN ↵ してみよう。どうなったかな？

画面には2個の数字が表示されたはずだ。左の数字は、マリオが方向転換するときのX座標値、右の数字はY座標値だ。

おもしろいかおもしろくないか　遊んでプログラム

たくさんのキャラクタがいろいろに動く

マリオとレディだけでは、あまりにも単純すぎるから、このほかにペンペンやらアキレスやら、いろいろなアニメキャラを加えて、いろいろに動くプログラムを載せておこう。

DEF MOVEやPOSITION文の数値を、変えて動かしてみてもおもしろいよ、きっと。

```
10 CLS
20 CGSET 1, 0
30 SPRITE ON
40 FOR M=0 TO 15
50 DEF MOVE (0) =SPRITE (M, 1, 1,
   255, 0, 0)
60 DEF MOVE (1) =SPRITE (M, 2, 1,
   255, 0, 0)
70 DEF MOVE (2) =SPRITE (M, 3, 1,
   255, 0, 0)
80 DEF MOVE (3) =SPRITE (M, 4, 1,
   255, 0, 0)
90 DEF MOVE (4) =SPRITE (M, 5, 1,
   255, 0, 0)
100 DEF MOVE (5) =SPRITE (M, 6, 1,
    255, 0, 0)
110 DEF MOVE (6) =SPRITE (M, 7, 1,
    255, 0, 0)
120 DEF MOVE (7) =SPRITE (M, 8, 1,
    255, 0, 0)
130 MOVE 0, 1, 2, 3, 4, 5, 6, 7
```

```
140  PAUSE 680
150  NEXT
160  FOR M=0 TO 15
170  DEF MOVE(0)=SPRITE(M, 1, 1,
     255, 0, 0)
180  DEF MOVE(1)=SPRITE(M, 2, 1,
     255, 0, 0)
190  DEF MOVE(2)=SPRITE(M, 3, 1,
     255, 0, 0)
200  DEF MOVE(3)=SPRITE(M, 4, 1,
     255, 0, 0)
210  DEF MOVE(4)=SPRITE(M, 5, 1,
     255, 0, 0)
220  DEF MOVE(5)=SPRITE(M, 6, 1,
     255, 0, 0)
230  DEF MOVE(6)=SPRITE(M, 7, 1,
     255, 0, 0)
240  DEF MOVE(7)=SPRITE(M, 8, 1,
     255, 0, 0)
250  A=RND(7)
260  B=RND(7)
270  C=RND(7)
280  D=RND(7)
290  POSITION A, 200, 100
300  POSITION B, 100, 140
310  POSITION C, 40, 40
320  POSITION D, 120, 40
330  MOVE 0, 1, 2, 3, 4, 5, 6, 7
340  PAUSE 680
350  NEXT
360  FOR M=1 TO 10
370  A1=RND(7)
380  DEF MOVE(0)=SPRITE(A1, 1,
```

```
     3, 255, 0, 0)
390  A2=RND(7)
400  DEF MOVE(1)=SPRITE(A2, 2,
     3, 255, 0, 0)
410  A3=RND(7)
420  DEF MOVE(2)=SPRITE(A3, 3,
     3, 255, 0, 0)
430  A4=RND(7)
440  DEF MOVE(3)=SPRITE(A4, 4,
     3, 255, 0, 0)
450  A5=RND(7)
460  DEF MOVE(4)=SPRITE(A5, 5,
     3, 255, 0, 0)
470  A6=RND(7)
480  DEF MOVE(5)=SPRITE(A6, 6,
     3, 255, 0, 0)
490  A7=RND(7)
500  DEF MOVE(6)=SPRITE(A7, 7,
     3, 255, 0, 0)
510  A8=RND(7)
520  DEF MOVE(7)=SPRITE(A8, 8,
     3, 255, 0, 0)
530  E=RND(7)
540  F=RND(7)
550  G=RND(7)
560  H=RND(7)
570  POSITION E, 20, 130
580  POSITION F, 100, 50
590  POSITION G, 140, 90
600  POSITION H, 210, 60
610  MOVE'0, 1, 2, 3, 4, 5, 6, 7
620  PAUSE 1080
630  NEXT
640  END
```

ステップ　2

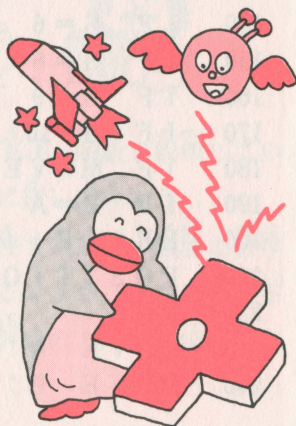
スターシップを画面に飛ばせ！

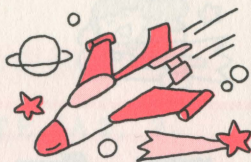
◆コントローラで動かせば自由にどこへでも飛んで行くよ

キャラクタテーブルAのアニメキャラを表示させたら、もっと自由に動かしてみようか。どうすればいいって？

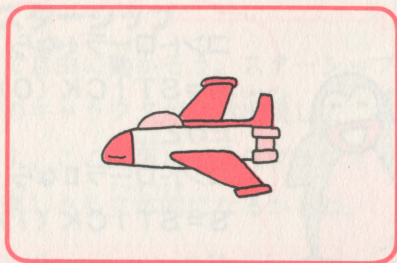
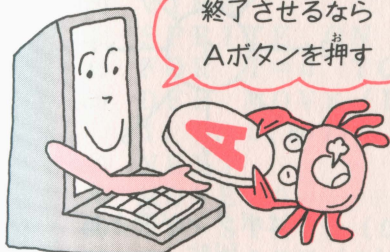
コントローラを使えば、アニメキャラはもっと自由自在に動かせるのだ。

いままでは、アニメキャラクタテーブルAにあるマリオやペンペン、レディなどのキャラを、ベーシックで画面に表示させたり、動かしたりしていたんだね。でも、どうもその動きが遅かったり、ぎこちなかったり、なんとなくものたりない感じがしなかったかい？

こんどは、表示させたアニメキャラの動きを、コントローラを使って操作しよ

っていうわけ。動きかたがもっと自由になるから、気分もサイコー自由になるかもね。

まず、次のプログラムを入力してからだ。

スターシップを自由に飛ばせる

```
10  CLS
20  SPRITE ON
30  FOR I=0 TO 7
40  DEF MOVE(I)=SPRITE(9, I+1,
    1, 1, 0, 0)
50  NEXT
60  MOVE 0
70  IF STRIG(0)=8 THEN END
80  S=STICK(0)
90  IF S=0 THEN 70
100  IF S=1 THEN A=2
110  IF S=2 THEN A=6
120  IF S=4 THEN A=4
130  IF S=5 THEN A=3
140  IF S=6 THEN A=5
150  IF S=8 THEN A=0
160  IF S=9 THEN A=1
170  IF S=10 THEN A=7
180  IF MOVE(B)=-1 THEN 180
190  IF B=A THEN 240
200  ERA B
210  POSITION A, XPOS(B), YPOS(B)
220  B=A
230  MOVE A:GOTO 70
240  MOVE B:GOTO 70
```

プログラムをRUNさせてみよう。画面のほぼ中央にスターシップが表示されたね。このスターシップを動かすのは、コントローラⅠだよ。

どんどん動かしてみてね。なにかがわかってくるはずだ。

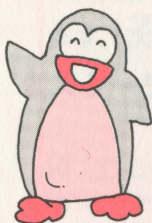
せっかく苦心して入力したプログラムだから、まずテープにセーブしておこうか。セーブのしかたは前に出ていたね。

十字形ボタンの押された方向調べ

次のプログラムを入力してみてね。

```
10  S=STICK (0)
20  PRINT  S
30  GOTO  10
```

このプログラムはなにかというと、STICK関数を説明しているんだ。十字形ボタンをいろいろな方向に押してみよう。

ボタンを押すたびに、画面に数字が表示されるはずだ。上なら８、下なら４だ。斜めの方向にも押してみよう。

コントローラⅠなら
S=STICK（0）
だけど
コントローラⅡなら
S=STICK（1）
だよ

つづいて短いプログラムをもうひとつ。入力したらいつものようにRUN ↵ だ。

```
10  T=STRIG（0）
20  PRINT T
30  GOTO  10
```

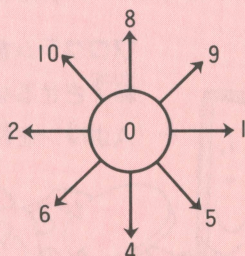
RUNさせたら、コントローラⅠのトリガボタンを押してみよう。前と同じように、どのボタンが押されたかで、違う数字が表示されるね。STRIG関数は、どのボタンが押されたかを調べるものなのだ。

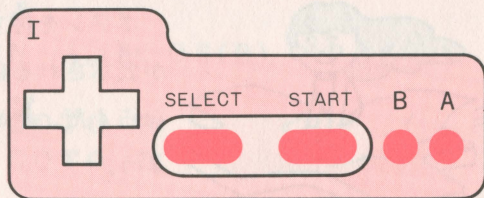
■**表示される数値**

START⟶1
SELECT→2
Bボタン⟶4
Aボタン⟶8

I
SELECT　START　B　A

＊十字形ボタンの方向

十字形ボタンが押された方向を数値で示すと次のようになる。

0はボタンが押されていない状態だ。

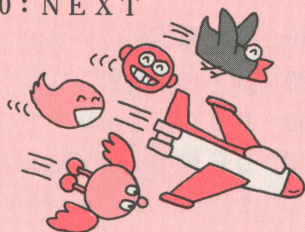
十字形ボタンの
押された方向は
前にもちょっと
出ているよ

キャラクタが出現。飛ぶスターシップ

スターシップをコントローラで自由に動かそう。スターシップが動いていないときは、７種類のキャラクタが突然出現して、右から左へ勝手に移動したりするよ。この画面に、ＢＧグラフィックで、夜空の背景でも描けば、楽しさも立体的になるかもね。

```
10  CLS:CGSET 1,0
20  VIEW
30  SPRITE ON
40  FOR I=0 TO 7
50  DEF MOVE(I)=SPRITE(9,I+1,
    1,1,0,0)
60  NEXT
70  MOVE 0
80  IF STRIG(0)=8 THEN END
90  S=STICK(0)
100  IF S=0 THEN 260
110  IF S=1 THEN A=2
120  IF S=2 THEN A=6
130  IF S=4 THEN A=4
140  IF S=5 THEN A=3
150  IF S=6 THEN A=5
160  IF S=8 THEN A=0
170  IF S=9 THEN A=1
180  IF S=10 THEN A=7
190  IF MOVE(B)=-1 THEN 190
200  IF B=A THEN 250
210  ERA B
220  POSITION A,XPOS(B),YPOS(B)
230  B=A
240  MOVE A:GOTO 80
250  MOVE B:GOTO 80
260  H=H+1
270  IF H>100 THEN 290
```

```
280  GOTO 80
290  Z=RND(7)+1
300  IF Z=1 THEN 370
310  IF Z=2 THEN 440
320  IF Z=3 THEN 510
330  IF Z=4 THEN 570
340  IF Z=5 THEN 630
350  IF Z=6 THEN 680
360  IF Z=7 THEN 730
370  DEF SPRITE 0,(0,1,0,0,0)=
     CHR$(64)+CHR$(65)+CHR$(66)+CH
     R$(67)
380  FOR N=255 TO 0 STEP -4
390  SPRITE 0,N,50
400  FOR W1=0 TO 100:NEXT
410  NEXT
420  H=0
430  GOTO 80
440  DEF SPRITE 1,(1,1,0,0,0)=
     CHR$(184)+CHR$(185)+CHR$(186)+
     CHR$(187)
450  FOR N=255 TO 0 STEP -4
460  SPRITE 1,N,100
470  FOR W2=0 TO 100:NEXT
480  NEXT
490  H=0
500  GOTO 80
510  DEF SPRITE 2,(0,1,0,0,0)=
     CHR$(96)+CHR$(97)+CHR$(98)+CH
     R$(99)
520  FOR N1=255 TO 0 STEP -4
530  SPRITE 2,N1,75
540  FOR W3=0 TO 100:NEXT
550  NEXT
560  H=0:GOTO 80
```

```
570  DEF SPRITE 3,(2,1,0,0,0)=
     CHR$(28)+CHR$(29)+CHR$(30)+CH
     R$(31)
580  FOR N=255 TO 0 STEP -4
590  SPRITE 3,N,125
600  FOR W4=0 TO 100:NEXT
610  NEXT
620  H=0:GOTO 80
630  DEF SPRITE 4,(2,1,0,0,0)=
     CHR$(192)+CHR$(193)+CHR$(194)+
     CHR$(195)
640  FOR N=255 TO 0 STEP -8
650  SPRITE 4,N,150
660  FOR W5=0 TO 100:NEXT
670  NEXT:H=0:GOTO 80
680  DEF SPRITE 5,(0,1,0,0,0)=
     CHR$(200)+CHR$(201)+CHR$(202)+
     CHR$(203)
690  FOR N2=255 TO 0 STEP -4
700  SPRITE 5,N2,175
710  FOR W6=0 TO 100:NEXT
720  NEXT:H=0:GOTO 80
730  DEF SPRITE 6,(0,1,0,0,0)=
     CHR$(88)+CHR$(89)+CHR$(90)+CH
     R$(91)
740  FOR N=255 TO 0 STEP -4
750  SPRITE 6,N,200
760  FOR W7=0 TO 80:NEXT
770  NEXT
780  H=0
790  GOTO 80
800  END
```

ステップ　3

デートしたいな マリオとレディ

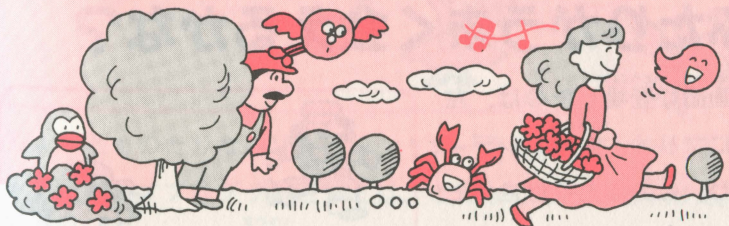
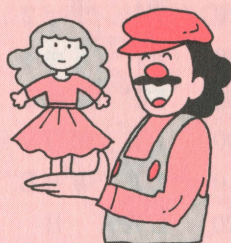
◆BGグラフィックで背景を描いたらもっと楽しい

アニメキャラクタテーブルAのキャラを画面に登場させて、コントローラIで動かす遊びのプログラムは、前で勉強したわけだけど、その応用で、マリオとレディをデートさせてみよう。

次のプログラムを入力したら、例によってRUN ⏎だよ。

●マリオとレディがデートした！

```
10　CLS：CGSET　1,0
20　SPRITE　ON
30　FOR　I＝0　TO　3
40　DEF　MOVE（I）＝SPRITE（0，I＊2＋
　　1，1，1，0，0）
```

```
50 NEXT
60 FOR I=0 TO 3
70 DEF MOVE(I+4)=SPRITE(1,I*
   2+1,1,1,0,2)
80 NEXT
90 POSITION 1,20,120
100 MOVE 1:B=1
110 POSITION 7,200,120
120 MOVE 7:D=7
130 IF STRIG(0)=8 OR STRIG(1)
    =8 THEN END
140 S=STICK(0)
150 IF S=0 THEN 270
160 IF S=1 THEN A=1
170 IF S=2 THEN A=3
180 IF S=4 THEN A=2
190 IF S=8 THEN A=0
200 IF MOVE(B)=-1 THEN 260
210 IF A=B THEN 260
220 ERA B
230 POSITION A,XPOS(B),YPOS(B)
```

```
240  B=A
250  MOVE B:GOTO 270
260  MOVE A
270  T=STICK(1)
280  IF T=0 THEN 130
290  IF T=1 THEN C=5
300  IF T=2 THEN C=7
310  IF T=4 THEN C=6
320  IF T=8 THEN C=4
330  IF MOVE(D)=-1 THEN 330
340  IF C=D THEN 390
350  ERA D
360  POSITION C,XPOS(D),YPOS(D)
370  D=C
380  MOVE C:GOTO 130
390  MOVE D:GOTO 130
```

ふたりがうまく会(あ)えるかな?

画面(がめん)に登場(とうじょう)したのは、左(ひだり)側(がわ)にマリオ、右側(みぎがわ)にレディだね。このふたりはきっと仲(なか)よしだよ。だけどまだ、

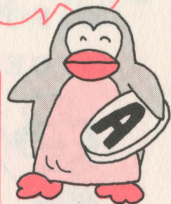
なんとなくふたりだけでデートしたことはないんだ。このふたりをせめて、画面の中でだけでもデートをさせてあげよう。

ここのマリオとレディは、ちょっと不器用にできていて、上下左右には自由に歩けるけど、斜めには動けないときている。そこらあたりはカンベンしてもらおう。

マリオを動かすのはコントローラⅠ、レディを歩かせるのはコントローラⅡだよ。

コントローラⅠは
マリオ
Ⅱはレディだよ

プログラムを
終わらせたいなら
どちらの
コントローラ
でもいいから
Aボタンを押す

あ、マリオ、横に行ったりして
斜めに行けば
レディのところまで近いのに

マリオもレディも
上下左右にしか
動けないんだから
不器用なものだ

ちょっとものたりないけど、コントローラを使っていろいろなキャラを動かすプログラムとしては、こんなところが基本形だ。簡単にプログラムの説明をしておこう。

■〝マリオとレディがデートした！〞プログラムの説明

BGグラフィックで
背景を描いて
それから
マリオとレディを
デートさせれば
もっと楽しいよ

30〜50行　どんなマリオをどう動かすかを決めている。マリオの動作番号は０〜３だ。

30　FOR　I＝0　TO　3……Iは0から3だよ

40　DEF　MOVE　（I）＝……Iは動作番号なのだ

60〜80行　どんなレディをどう動かすかを決めている。

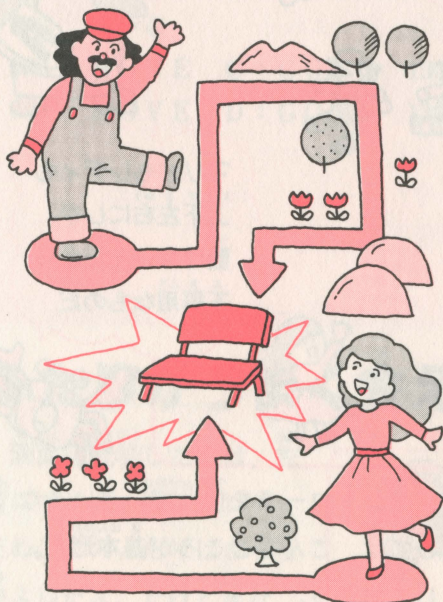
60　FOR　I＝0　TO　3……Iは0から3だけど……

70　DEF　MOVE　（I＋4）……動作番号はIに4を加えるから4〜7だ。

90〜120行　マリオとレディを最初に表示させる位置を決めて。

130行　どちらかのコントローラのAボタンが押されたら、プログラムは終了する。

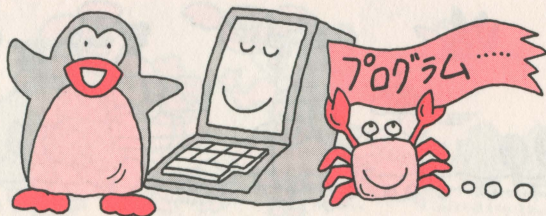
140〜260行　マリオを移動させるきまりを作る。

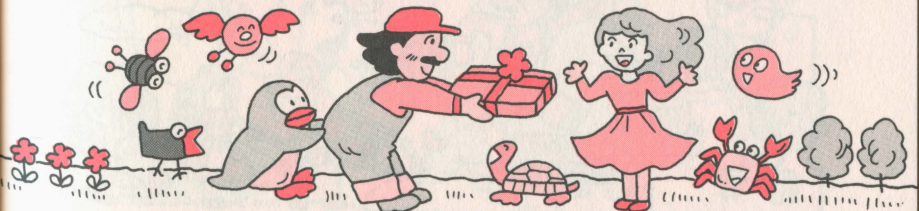
270〜390行　レディを移動させるきまりを作る。

おもしろいかおもしろくないか　**遊んでプログラム**

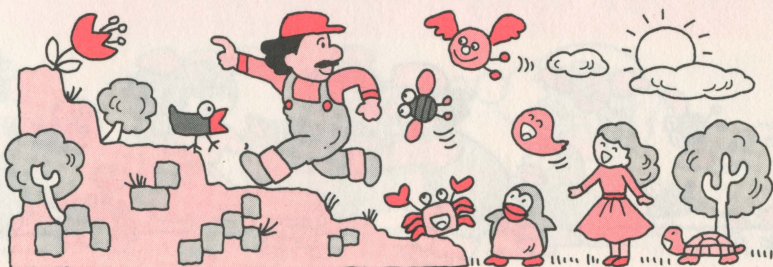
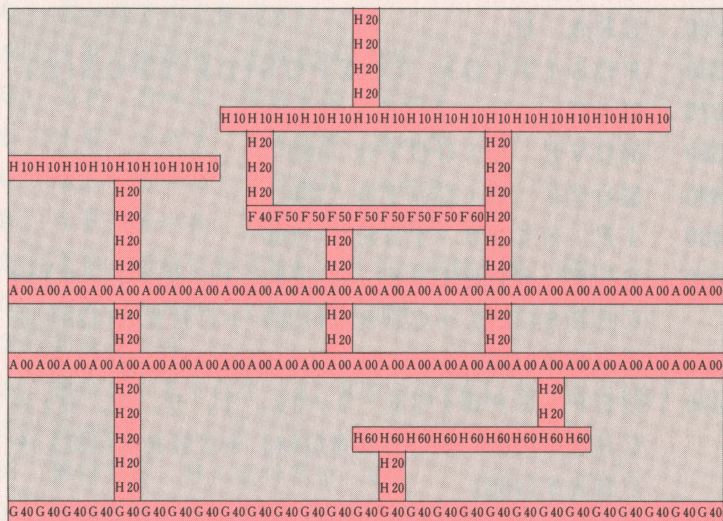
●背景のある楽しいマリオとレディのデート

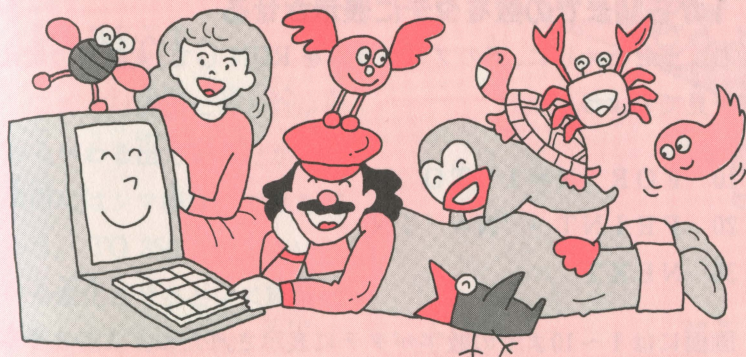
デートするときには、まわりに公園の木があったり、遠くの山がかすんでいたりと、なにかいろいろな背景に囲まれていたほうが、画面も楽しくなるね。ファミコンのBGグラフィックを使えば、背景が描けるから、その背景の中でふたりをデートさせてみよう。

```
10 CLS:CGSET 1,0
20 SPRITE ON
25 VIEW
30 FOR I=0 TO 3
40 DEF MOVE(I)=SPRITE(0,I*2+
   1,1,1,0,0)
50 NEXT
60 FOR I=0 TO 3
70 DEF MOVE(I+4)=SPRITE(1,I*
```

プログラムの最初は
前と同じだから
新たに入力しなくても
前のを利用すればいい

```
 2+1, 1, 1, 0, 2)
80 NEXT
90 POSITION 1, 20, 120
100 MOVE 1:B=1
110 POSITION 7, 200, 120
120 MOVE 7:D=7
130 IF STRIG(0)=8 OR STRIG(1)
 =8 THEN END
140 S=STICK(0)
150 IF S=0 THEN 270
160 IF S=1 THEN A=1
170 IF S=2 THEN A=3
180 IF S=4 THEN A=2
190 IF S=8 THEN A=0
200 IF MOVE(B)=-1 THEN 260
210 IF A=B THEN 260
220 ERA B
230 POSITION A, XPOS(B), YPOS(B)
240 B=A
250 MOVE B:GOTO 270
260 MOVE A
```

```
270  T=STICK(1)
280  IF T=0 THEN 400
290  IF T=1 THEN C=5
300  IF T=2 THEN C=7
310  IF T=4 THEN C=6
320  IF T=8 THEN C=4
330  IF MOVE(D)=-1 GOTO 330
340  IF C=D THEN 390
350  ERA D
360  POSITION C,XPOS(D),YPOS(D)
370  D=C
380  MOVE C:GOTO 130
390  MOVE D:GOTO 130
400  IF S<>0 THEN 140
410  DEF SPRITE 0,(0,1,0,1,0)=
     CHR$(121)+CHR$(120)+CHR$(123)+
     CHR$(122)
420  DEF SPRITE 1,(0,1,0,1,0)=
     CHR$(165)+CHR$(164)+CHR$(167)+
     CHR$(166)
430  FOR N=1 TO 240
```

```
440   SPRITE  0, N, 130
450   NEXT
460   FOR  N=1  TO  240
470   SPRITE  1, N, 150
480   NEXT
490   GOTO  140
```

背景(はいけい)を作(つく)るBGグラフィックのパターン図(ず)

パート3

キャラクタと自由に遊ぼう

マリオが行ったり来たり

♣FOR～NEXT命令は同じ動作をくり返させる

ここでは、FOR～NEXTの勉強をするわけだけど、FOR～NEXTはFORとNEXTの間をTOのあとに指定した数値の回数だけくり返すんだ。

```
FOR  N=1  TO  10
  ～          Nが1から10になるまで10回くり返す
NEXT
```

●1から10までの数をタテに表示させる

ちょっとためしに、下のプログラムをRUNして、変数Nの増え方を見てみよう。

```
10  FOR  N=1  TO  10
20  PRINT ,"N= ";N
30  NEXT
```

画面には1～10までの数字がタテに表示されたね。このような

働きをするのが、FOR～NEXT なんだ。

このFOR～NEXT の使い方は、同じことをさせるときには、とても便利な命令なんだよ。

●画面全体を青い■で塗りつぶす

たとえば、何かキャラクタを画面いっぱいに表示したい場合には、どうしようか？

```
10  FOR  N=1  TO  700
20  PRINT  "■";
30  NEXT
```

■の出し方はカナモードでGRPHキーと¥ルキーを押すんだよ

このプログラムをRUN すると、画面全体が青色になるよ。その意味は"■"をコンピュータが、いっしょうけんめいに書いているからなんだ。

●青と白のシマ模様を表示させよう

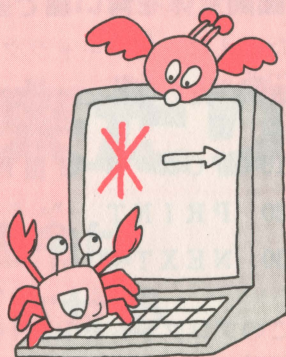
```
10  FOR  N=1  TO  350
20  PRINT  "■□";
30  NEXT
```

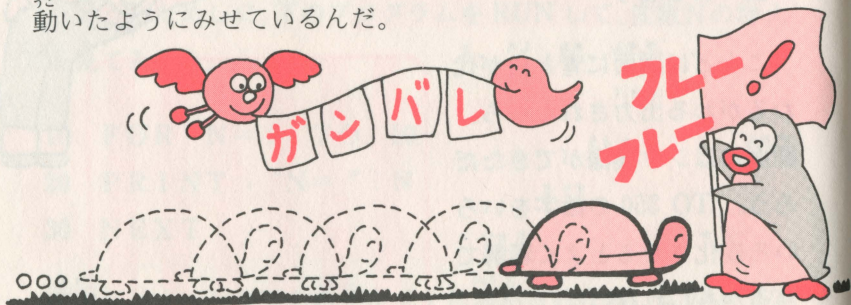
□の出し方はカナモードでGRPHキーとへリキーを押すんだよ

こんどは画面に青と白が代わるがわる出力されたから、画面にはシマ模様ができただろう。TO 350 の数字をいろいろ変化させるとよく理解できるかもしれないな。

●＊マークを左から右に動かそう

こんどはもう少しおもしろくしてみよう。キャラクタを左から右に走らせるプログラムだ。少し複雑だから、注意してね。

```
 5 CLS
10 FOR N=1 TO 27
20 LOCATE N, 10
30 PRINT "＊"
40 FOR T=1 TO 100
50 NEXT
60 LOCATE N, 10
70 PRINT" "
80 NEXT
```

書いて消して動いたようにみせる

これをRUNすると、＊マークが右に移動したはずだけど、どうしてかわかるかな？　＊マークが動いたようにみえるけど、実は、最初に書いた＊マークを消すと同時に、その右に＊マークを書き、書いて消して、書いて消して……と同じ動作を何回もくり返して、動いたようにみせているんだ。

＊カウンターループ

40行と50行のFOR〜NEXTは、カウンターループといって、少し時間をかけているところだよ。40行の100の数字を少なくすれば早く動き、多くすればゆっくりすすむよ。

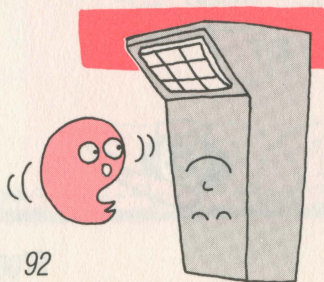
マリオをFOR〜NEXTで右に動かす

ここでいよいよマリオの登場といこう。マリオを左から右へ移動させるには、MOVE命令のほかに、FOR〜NEXTを使ってもいい。

```
10 CLS
20 SPRITE ON
30 CGSET 1, 0
40 DEF SPRITE 0, (0, 1, 0, 1, 0) =
   CHR$(1) +CHR$(0) +CHR$(3) +C
   HR$(2)
50 FOR X=0 TO 244 STEP 4
60 SPRITE 0, X, 100
70 FOR M=0 TO 100:NEXT
80 NEXT
90 END
```

左右逆のマリオが走るぞ！

このプログラムをRUNすると、右を向いたマリオが左から右方向へ走って行く。40行を見てみよう。ところがキャラクタテーブルにはどこをさがしても、右向きのマリオはないじゃないか。そのキャラクタを右に向かせて登録しているのがこの40行だよ。

SPRITE　0のあと、カッコ内4番目の数字が0なら左向きのそのままのキャラクタ表示、1なら右向き、つまり逆向きのキャラクタが表示できるんだ。でもこれだけではだめ。その後ろにあるCHR$の数字も逆に設定しなければならないんだよ。

＊アニメキャラクタの表示

40行のDEF SPRITE 0は、表示するキャラクタ名を〝0〟とするということだが、これにつづく(　)の中の意味を教えよう。

（0，1，0，1，0）

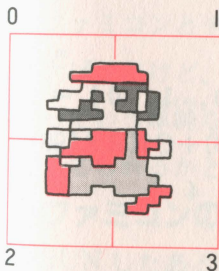
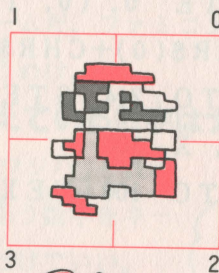
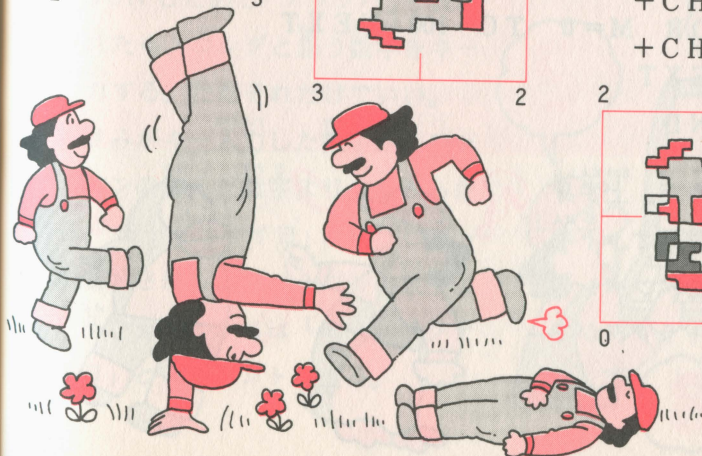
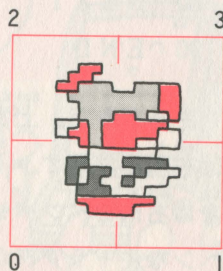
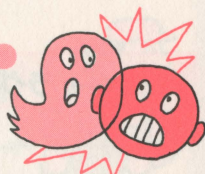
- 5番目の0：アニメキャラクタを上下逆に表示したかったら1
- 4番目の1：1は左右を逆にして表示。そのままの表示なら0
- 3番目の0：スプライト面のキャラクタがバックグラウンド面の文字の上に表示。1なら文字が上
- 2番目の1：4文字分の大きさで表示。0なら1文字分の大きさ
- 1番目の0：キャラクタテーブルと同じ配色

■キャラクタの逆立ちや裏返しの表示

キャラクタテーブルAにはいろいろなキャラクタがあって、どれでも自由に表示させることができるけど、このキャラクタを逆立ち（上下逆の表示）させたり、裏返し（左右逆の表示）させたりすることもできる。

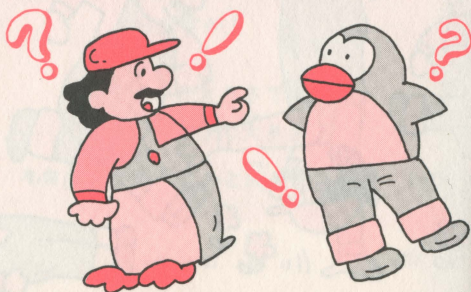
前のプログラムの40行をいろいろ変えてためしてみよう。

プログラムの40行
DEF SPRITE 0,(0,1,0,1,0)
=のあとをいろいろ変化させるのだ

（正常な表示）＝CHR＄(0)+CHR＄(1)+CHR＄(2)+CHR＄(3)

（左右逆の表示）＝CHR＄(1)+CHR＄(0)+CHR＄(3)+CHR＄(2)

（上下逆の表示）＝CHR＄(2)+CHR＄(3)+CHR＄(0)+CHR＄(1)

●2つのキャラクタをドッキング

キャラクタテーブルAのマリオもレデイもペンペンも、そのなかのキャラクタも、逆立ちさせたり、左右逆に表示させたり、自由に表示させられるようになったはずだけど、この方法を応用すれば、もっとおもしろいこともできそうだ。

たとえば、上半身がマリオで下半身がペンペンのヘンな生きものだって表示させられるのだ。2つ以上のキャラクタをドッキングさせて、新しいキャラクタを創造してみよう。

```
10 CLS
20 SPRITE ON
30 CGSET 1，0
40 DEF SPRITE 0，(0，1，0，1，0)＝
   CHR$(1)＋CHR$(0)＋CHR$(99)＋CHR$(98)
50 FOR X＝0 TO 244 STEP 4
60 SPRITE 0，X，100
70 FOR M＝0 TO 100：NEXT
80 NEXT
90 END
```

さあどうだ。上半身は右向きマリオ、下半身は右向きペンペンのヘンなキャラクタができたぞ。

ステップ　2

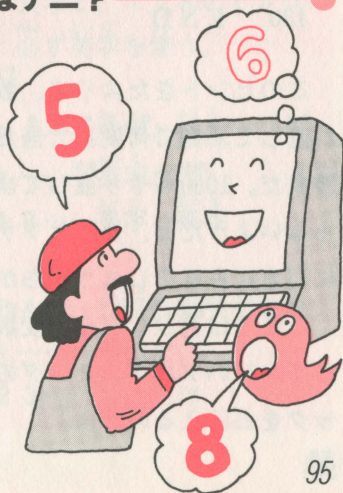
数当てゲームでファミコンに挑戦

♣文字変数も数値変数も変数はいろいろ変わる数だ

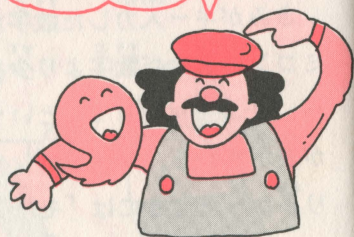
ファミコンとチエくらべしてみようか。チエくらべとはちょっと大げさだけど、ファミコンが選んだ1～99までの数字を当てようっていう単純なゲームだ。ファミコンが選んだ数字を、何度目で当てられるかというわけ。

●ファミコンの考えている数字はナニ？

RUN したあと「スウジハ」と表示されたら、コレダと思う数字をキー入力する。ただそれだけでいい。

きみがキー入力した数字が、ファミコンの選んだ数字より少なかったら、「モットオオイヨ」というヒントが表示される。ファミコンの数字より多かったときには「モットスクナイヨ」というヒントが出る。

```
10 CLS
20 A=RND (99)
30 KI=1
40 PRINT KI;"カイメ "
50 INPUT "スウジ ハ=";B
60 IF A=B THEN PRINT "アタリ"
   :GOTO 100
70 KI=KI+1
80 IF A>B THEN PRINT "モット オオ
   イヨ":GOTO 40
90 IF A<B THEN PRINT "モット スク
   ナイヨ":GOTO 40
100 PRINT "モウイチド シマスカ (Y/N)"
110 INPUT CY$
120 IF CY$="Y" THEN GOTO 10
130 END
```

このヒントをたよりに、数字を入れ直して、さて何度目で当たるかが勝負だ。10回もやり直してまだ当たらないようだと、きっとファミコンにバカにされてしまうにちがいない。

ここでは遊びながら、変数というむずかしいプログラミングのテクニックをおぼえるのだよ。

変数はいろいろなものを入れる箱

プログラムの20行を見てみよう。

20　A＝RND(99)

これは、Aという箱の中に、0～98までのどんな数でも1つ選んで入れなさいという文だ。いい換えれば、Aは、0～98までのどんな数にでも変わるよ、というわけで、このAが変数なのだ。

変数には2種類ある。数値変数と文字変数(ストリング変数)だ。上の例で、Aという変数は0～98までの数値に変わるのだから、これは数値変数だ。

＊数値変数の作り方

数値変数を作るときの約束ごとをおぼえよう。

1．使えるのは、数値と英字だけ。

2．1文字か2文字でつける。

3．2文字のときの最初の文字と、1文字のときは必ず英字を使う。

(正しい例)

A1……英字と数値

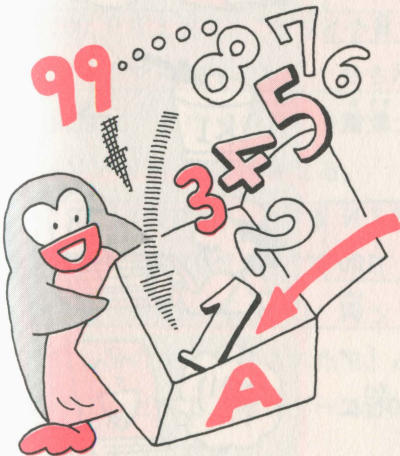
A………英字だけ

AB……英字と英字

(まちがった例)

2A……最初の文字が数値

1………数値だけ

23……数値だけ

●数当てゲームのプログラムを読む

数値変数は、いろいろな数値を入れる箱だったね。それでは文字変数はいったいなんだろう？　いろいろな文字を入れる箱とでもいっておこうか。

文字変数については、あとでくわしく説明することにして、まず数当てゲームのプログラムを、最初からかんたんに説明しておこう。

```
10 CLS
```

表示させる画面をきれいにそうじだ。

```
20 A=RND (99)
```

数値変数Aに、0～98までの数値のうち、なにかを代入する。ここでファミコンの選んだ数値が23だとすれば、変数Aには23が代入される。

```
30 KI=1
```

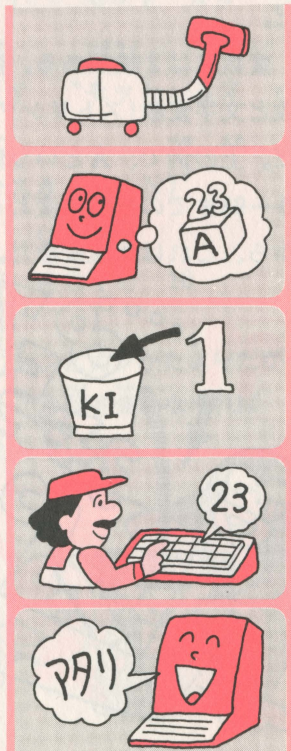
最初のゲームだから、変数KIには1を代入する。

```
40 PRINT KI;"カイメ  ";
50 INPUT "スウジ ハ=";B
```

いまのゲームが何回目かを表示させたあと、きみたちがキー入力した数値を変数Bに入れる。

```
60 IF A=B THEN PRIN
   T "アタリ":GOTO 100
```

ファミコンが考えた数値（A）と入力した数値（B）が同じだったら「アタリ」と表示してゲーム終了。100行に飛ぶ。

```
70 KI=KI+1
```

１回目でまちがったひとは、回数の変数ＫＩに＋１する。

```
80 IF A>B THEN PRINT "モット オオイヨ":GOTO 40
```

入力した数値(Ｂ)が、ファミコンの選んだ数値(Ａ)より小さかったら、「モット　オオイヨ」と表示して40行へ。ゲームはやり直しだ。

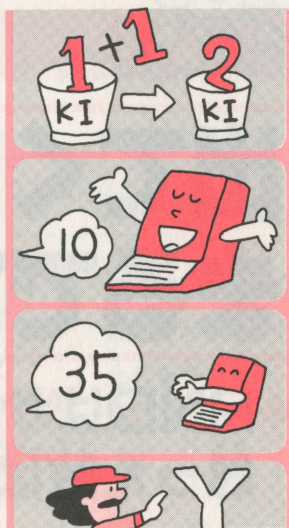
```
90 IF A<B THEN PRINT "モット スクナイヨ":GOTO 40
```

入力した数値がファミコンの選んだ数値より大きかったら、「モット　スクナイヨ」と表示して40行へ。ゲームはやり直し。

```
100 PRINT "モウイチド シマスカ(Y/N)"
```

画面に「モウイチド　シマスカ(Ｙ／Ｎ)」と表示させる。

```
110 INPUT CY$
120 IF CY$="Y" THEN GOTO 10
```

もしＹのキーが押されたら、10行にもどって、再びゲームができるようになる。

＊ IF～THEN

もし雨だったらファミコンしよう、というように、もし～だったら、と条件をつけるのがＩＦ～THEN文だ。

60行のＩＦ　Ａ＝Ｂ THEN…は、ＡとＢが同じだったら

80行のＩＦ　Ａ＞Ｂは、ＡがＢより大きかったら

90行のＩＦ　Ａ＜Ｂは、ＡがＢより小さかったらという意味だ。

出てくる数はまるででたらめだ

♣RNDはサイコロの目だ 思った数はなかなか出ない

ファミコンは、でたらめな数を、でたらめな順番で発生させたり表示させたりすることができるよ。

数当てゲームのプログラムにも、

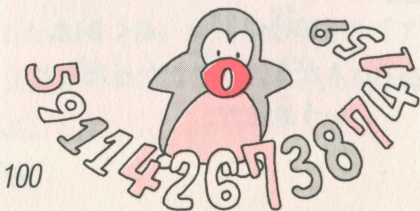
20　A＝RND(99)

というのがあったね。この意味は、0～98までの数のうち、どれでも勝手に変数Aに代入せよ、ということだつた。

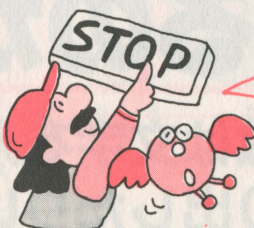
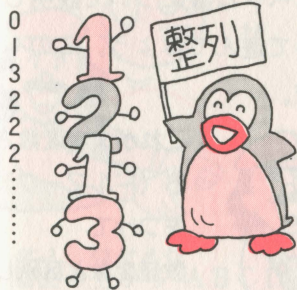
ここに使ったRNDが、でたらめな数をでたらめに発生させるもとなのだ。ランダム関数っていうんだよ。

●でたらめな数をでたらめに表示させる

RND（ランダム）関数を使った短い短いプログラムで、でたらめな数の発生をたしかめてみよう。

```
10　A＝RND（4）
20　PRINT　A；
30　GOTO　10
```

このプログラムをRUNさせてみよう。

0から3までの数が、でたらめの順序で、画面の左から右にどんどん表示されていくね。

1302331002312………
2030011………………

こんどは、20行の最後にある；(セミコロン)をつけないで、

```
20　PRINT A
```

と、こうしてRUNさせてみよう。どうなったかな？

さっきは、でたらめな数が、画面の左から右へ、どんどん表示されていったけれど、こんどは、画面の左上から、どんどん下に表示されていくだけだね。

；（セミコロン）があるかないかで、こんなにちがうんだよ。

＊RND関数

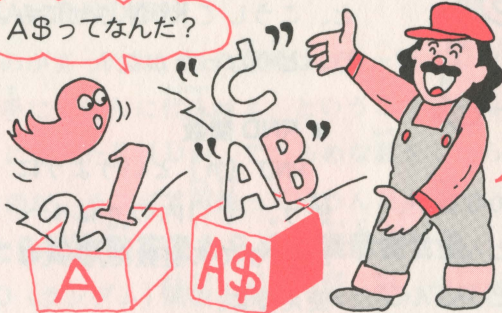
RND（n）というように使う。（　）内のnは1～32767までの整数だ。数式も使えるよ。

RND（5）なら、0～4までの数、RND（10）なら、0～9までの数を発生させるんだ。

文字のたし算はできるのかな

♣文字変数は文字を入れる箱 入れたり出したりいそがしい

ステップ3では、数値変数っていうヘンな数について説明したが、ここでは、もっとヘンな数を使って遊んでみよう。このヘンなヘンな数は、文字変数っていうんだ。

数値変数は、とりあえず数を入れたり出したりする箱だといって説明したが、これにならって、文字変数は、文字や文字列を入れたり出したりする箱だといっておこう。

まずは、次のプログラムを入力して、RUN させてみよう。

●文字変数を使って文字列を表示させよう

```
10  A$="ABCD"
20  B$="EFGH"
30  PRINT  A$
40  PRINT  B$
50  C$=A$+B$
60  PRINT  C$
70  END
```

さて、画面にはどんな表示が出たかな？

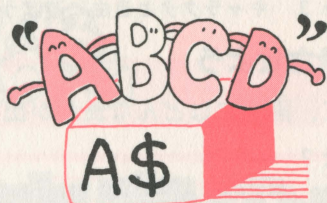
```
ABCD
EFGH
ABCDEFGH
```

画面の上から「ABCD」、その下に「EFGH」、さらにその下に「ABCDEFGH」と表示されたはずだ。

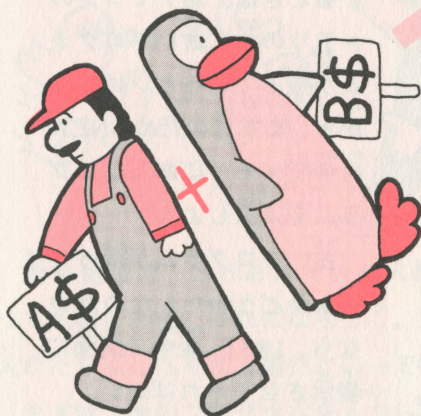
プログラムをざっと見てほしい。10行にA$、20行にB$、50行にC$という字が出てくるね。これを文字変数というのだ。

＊新しいプログラムの入力

新しいプログラムを入力するときは、ファミコンのメモリの中に古いプログラムが残っているといけないから、まず念のため、NEW命令でメモリにあるプログラムを消去しよう。

古いプログラムが残っているかどうかたしかめたいなら、LIST 命令で画面に表示させてみればいい。

10行のＡ＄＝〝ABCD〞は、Ａ＄という文字変数（文字を入れる箱に、〝〞（ダブルクォーテーション）で囲まれたABCDという文字列を入れる（代入する）という意味だ。

30行のPRINT A$で、文字変数Ａ＄に代入されたＡＢＣＤを表示させている。

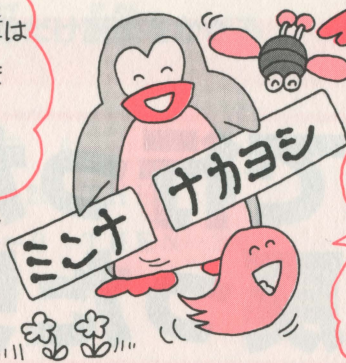
20行のＢ＄＝〝EFGH〞も同じだ。40行のPRINT文で表示される。

＊文字変数と変数値

文字変数は数値変数と区別するために、変数名のあとに＄（ドルマーク）をつける。

Ａ＄、Ｂ＄、Ｃ＄と、みんな＄がついているね。

また、文字変数に入れるデータ（変数値）は、〝〞（ダブルクォーテーション）で囲む。

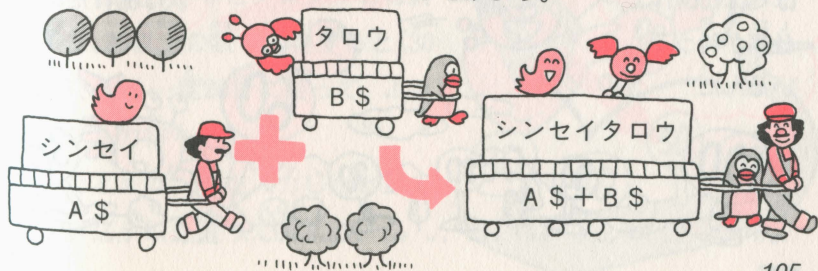
〝ABCD〞　〝EFGH〞と、こういうぐあいだ。

50行のＣ＄＝Ａ＄＋Ｂ＄は文字列のたし算だ。Ｃ＄という文字変数に、Ａ＄のABCDとＢ＄のEFGHを代入するわけだから、Ａ＄とＢ＄が並んだABCDEFGHが表示されている。

■自分の名まえを文字変数に入れる

こんどは、きみの名まえを入れてみようか。きみの名まえが、新星太郎だとして、前のプログラムをそっくりいただいてしまえ。

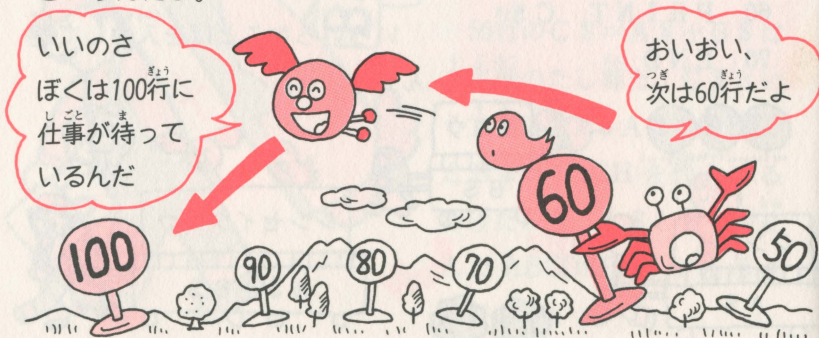
```
10　A＄＝〝シンセイ〟
20　B＄＝〝タロウ〟
30　PRINT　A＄
40　PRINT　B＄
50　C＄＝A＄＋B＄
60　PRINT　C＄
70　END
```

```
シンセイ
タロウ
シンセイタロウ
```

いろいろなことばを入力して遊んでみよう。

飛んで行ったり　また戻ったり

♣GOTOやGOSUBは流れを変えるジャンプ命令

プログラムは、行番号の小さな順に実行していく、というタテマエはもう知っているね。だけど、例外はどこにでもあるものだ。GOTO命令やGOSUB命令もそのひとつだ。

行番号の小さい順から実行していくどころか、そんなきまりは守ろうという気もさらさらなく、行きたい行番号のところに飛んで行ってしまう。だから、GOTOやGOSUB命令をジャンプ命令というんだよ。

＊が好きな数だけ画面をかざる

```
10  CLS
20  INPUT A
30  IF A>20 THEN 20
40  IF A=0 THEN 200
50  GOSUB 100
60  GOTO 20
100  FOR N=1 TO A
110  PRINT "＊";
120  NEXT
130  RETURN
200  END
```

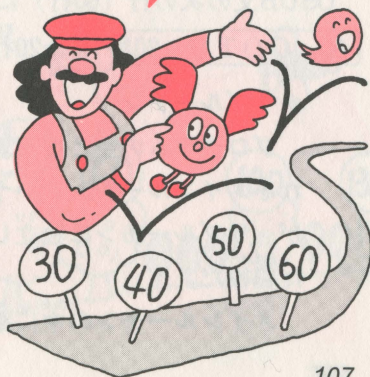
20行に戻る
100行にジャンプ
60行に戻る
サブルーチン

このプログラムは、数字キーで入力する20までの任意の数だけ、画面に「＊」を表示させるんだ。

RUN させると、20行の INPUT 文でキー入力待ちになる。１から20までの数をキー入力しよう。

０を入力するとプログラムは終了だ。20以上の数を入力すると(30行)、もう一度20行に戻って、ふたたび入力待ちだ。

たとえば、５を入力したとして、プログラムを追ってみよう。

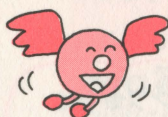
ここに出てくるジャンプ命令は
30行、40行のＩＦ～ＴＨＥＮ
50行のGOSUB
60行のGOTOだ

10 CLS …画面をきれいにして、

20 INPUT A …このAに5が入る。

30 IF A>20 THEN 20 …入力した数が20以上だったら ~ という条件だから、この行は無視。

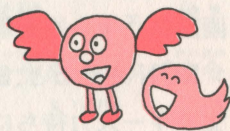
40 IF A=0 THEN 200 …入力した数は0ではないから、この行も無視。

50 GOSUB 100 …100行へジャンプしろというから、100行に行ってみよう。

100 FOR N=1 TO A …ここからサブルーチンに入る。Aは5だから、次のことを5回くり返す。

110 PRINT "*"; …「*」を5回表示するということになる。

120 NEXT …100行のFOR文を受けてFOR~NEXTの終了。

130 RETURN …50行の GOSUB文を受けて、GOSUB~RETURNの終了。GOSUB文の次の行（60行）に戻る。

60 GOTO 20 …また20行に戻って、もう一度くり返し。

＊サブルーチン

プログラムの主となる流れをメインルーチンというが、サブルーチンは独立した脇役だ。

メインルーチンの中に割り込んで、ひとつの役割を果たしたあとで、またメインルーチンに戻る。

このプログラムでは、100行からのFOR~NEXT文がサブルーチンだよ。

■ジャンプ命令はたくさんあるよ

＊を表示させるプログラムには、ジャンプ命令がたくさん出てきたよ。どれがジャンプ命令だろう？

■ GOTO 命令

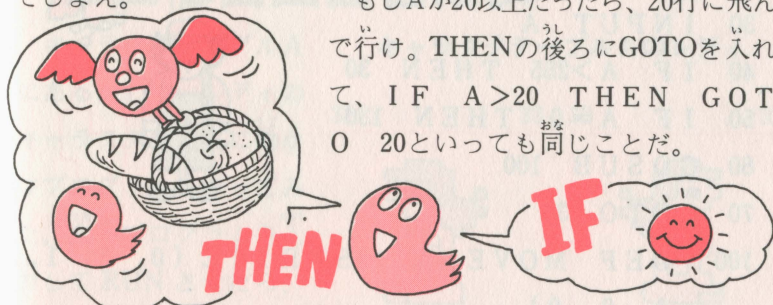
60　G O T O　20

20行に飛んで行け。

行番号の順序なんてほっといて、なにがなんでも20行にジャンプしてしまえ。

■ IF～THEN 命令

30　I F　A＞20　T H E N　20

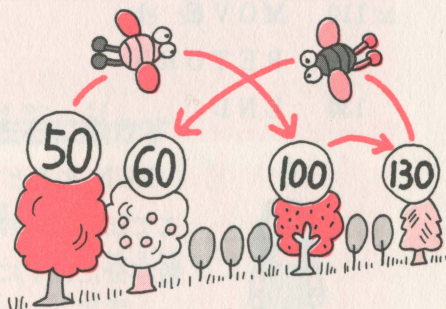
もしAが20以上だったら、20行に飛んで行け。THENの後ろにGOTOを入れて、I F　A＞20　T H E N　G O T O　20といっても同じことだ。

■ GOSUB～RETURN 命令

50　G O S U B　100

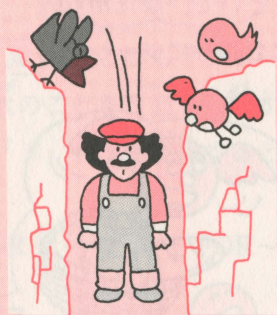
～

130　R E T U R N

まず100行にジャンプして、RETURN文に出会ったら前に戻って、GOSUB文の次の行に行けという命令だ。

●マリオがどんどん下に下がっていく

前のプログラムで、GOTOやGOSUB~RETURNのジャンプ命令を簡単に説明したけど、プログラムを実行しても、ただ「*」が表示されるだけで、ちっともおもしろくもなかったね。

こんどは、あのプログラムを応用して、少しは変化のあるものを作ってみた。キャラクタテーブルAにあるマリオのキャラクタを表示させ、その表示位置を、きみが操作する数字キーで変えてみようというわけだ。

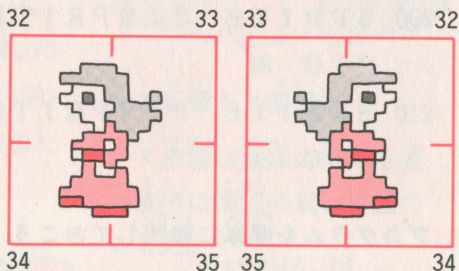
```
10  CLS
20  SPRITE ON
30  INPUT A
40  IF A>255 THEN 30
50  IF A=0 THEN 130
60  GOSUB 100
70  GOTO 30
100  DEF MOVE (1) =SPRITE (0, 5, 1,
     A, 0, 0)
110  MOVE 1
120  RETURN
130  END
```

RUNするとすぐ入力待ちの状態になる。そこで255以下の数字をキー入力してみる。押した数字分だけ下に、マリオが移動するよ。もし、0を入力すれば、プログラムは終了する。

ステップ　6

左右に動くかわいいレディ

♣キャラクタの表示や移動も自由自在だよ

キャラクタテーブルAにあるかわいいレディのキャラクタを左右に動かすプログラムを作ってみた。次のプログラムを入力してRUNさせたら、レディを動かすのは、コントローラIの十字形ボタン ← と → だよ。

キャラクタテーブルAのレディ

●コントローラでレディを左右に動かす

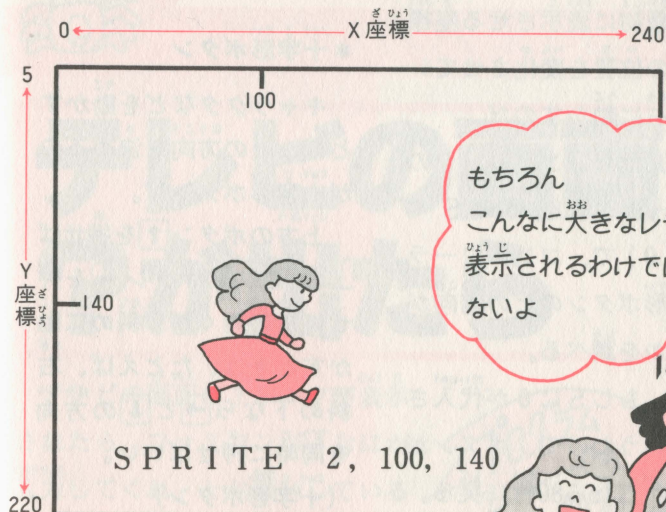
```
10 CLS
20 SPRITE ON
30 CGSET 1, 0
```

```
40 DEF SPRITE 2,(2,1,0,1,0)=
CHR$(33)+CHR$(32)+CHR$(35)+CHR
$(34)
50 DEF SPRITE 3,(2,1,0,0,0)=
CHR$(32)+CHR$(33)+CHR$(34)+CHR$(35)
60 SPRITE 2,100,140
70 X=100:Y=140
80 S=STICK(0)
90 IF S=0 THEN 80
100 IF S=1 THEN X=X+4:GOTO 200
110 IF S=2 THEN X=X-4:GOTO 210
120 GOTO 80
200 SPRITE 3:SPRITE 2,X,Y:G
OTO 80
210 SPRITE 2:SPRITE 3,X,Y:G
OTO 80
```

■**プログラムを簡単に説明しておこう**

10〜30行は初期設定だ。キャラクタテーブルを動かすときのきまり文句みたいなものだから、このまま暗記すればいい。

40行と50行は、表示させるレディのキャラクタを決めている。スプライト2のキャラクタは左右を逆にして右を向いたレディ、スプライト3は、キャラクタテーブルにあ

表示させる
キャラクタを
決めるのは
DEF SPRITE
だ

るままの左向きレディなのだ。

60行はSPRITE 2で定義した右向きのレディを、X座標を100、Y座標を140の位置に表示するというわけだ。

＊X座標とY座標

X座標は画面の横の位置、Y座標は画面の縦の位置のことだ。

スプライト画面上の有効表示範囲は、横（X座標）0～240、縦（Y座標）5～220だ。この範囲内だったら、レディは自由に動けるが、この範囲外を指定するとエラーになってしまうよ。

70行は最初に表示させる座標位置。この位置を変化させてレディを左右に動かすのだ。

80～120行が、コントローラの入力ルーチン。80行のS＝STICK（0）で、コントローラIの十字形ボタンのどの方向が押されたかを調べる。

90行は、もしSに0が代入されたらというわけで、ボタンが押されなかったら80行に戻る。

100行は、もしSに1（十字形ボタンの右方向）が代入されたら右に動くといっている。

110行は、もしSに2（十字形のボタンの左方向）が代入されたら、左に動くといっている。

200行で右向きのレディ（SPRITE　2）、210行なら左向きのレディ（SPRITE　3）を表示させて、右と左に動かす。

コントローラIIを
調べるときは
STICK（1）だ

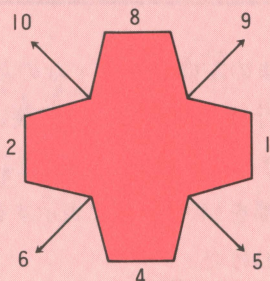
＊十字形ボタン

キャラクタなどを動かすとき、その方向を決めるのが十字形ボタンだ。

上方のボタン[↑]を押せば上に、左方のボタン[←]を押せば左に動くが、斜めに動かすときは、たとえば、右斜め下なら[→]と[↓]の方向を同時に押せばいい。

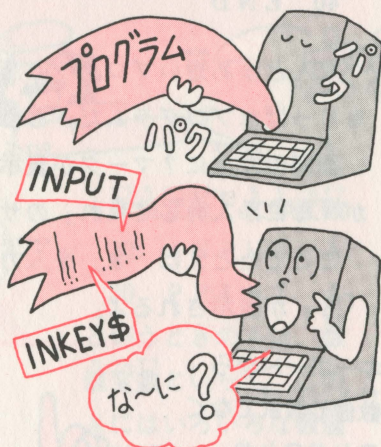
（十字形ボタン）

ボタンが押されたそれぞれの方向には、ファミコンがおぼえている決まった数字がついている。上なら8、下なら4だ。あとで出てくるから、おぼえておこう。

ステップ　7

テレビの画面に?が出たら

テレビの画面に？マークが表示されたら、ファミコンが「なにか入力してくれ」とお願いしているのだから、キーボードかあるいはコントローラを使って、なにか入力してあげなければいけない。

それでは、なにかを入力する命令には、どんなものがあるかな。少しくわしく説明しておこう。

ファミコンになにかを入力してやるときの命令には、INPUT、INKEY$、STICK、STRIG などがある。これらの使い方をおぼえておこうか。少しおもしろ味がないかもしれないが、だいじなことだから、がまんするんだね。

ファミコンは声は出さないけど、かわりに？マークを出して、きみに話しかけるよ

♣INPUTで入力したらPRINTで表示させてみる

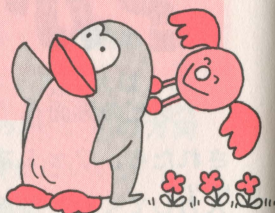
次のプログラムを入力して、RUNさせてみよう。

```
10 CLS
20 INPUT A      RUN↵
30 PRINT A     →
40 END
```

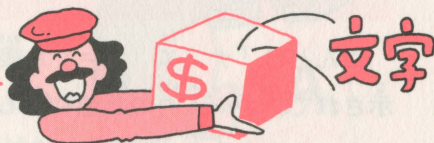
```
?
■
```

なんとまあ短いプログラムだ。でも短くともりっぱなプログラムだ。どうなったかな？

画面の左上に？マークが表示されて、カーソルもあるね。これが「なにか入力してくれ」のサインなのだ。

ここでたとえば「8」を入力してみよう。画面にいま入力した「8」が表示されるよ。

こんどはちょっと変えて、次のプログラムの実行だ。

```
10  CLS
20  INPUT  A$
30  PRINT  A$
40  END
```

こんどは、キーボードから文字を入れてみよう。EでもEFでもEFGでもいいよ。入力した文字がそのまま表示されたはずだ。

＊数値変数と文字変数

```
20  INPUT  A
30  PRINT  A
```

というときのAは、数値変数といって、このAにはいろいろな数値や数式があてはめられる。

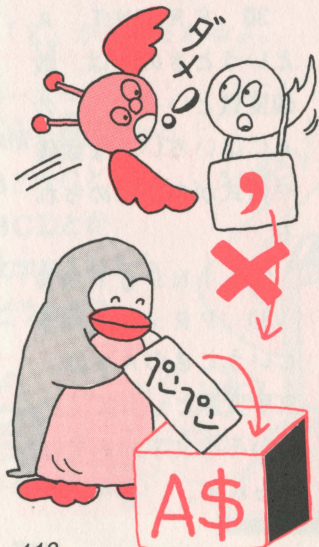
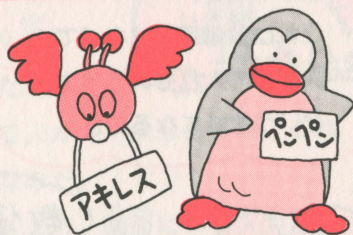
```
20  INPUT A$
30  PRINT A$
```

というときのA$は、文字変数といって、いろいろな文字があてはめられる。

●メッセージをつけておくと便利だよ

ただ「？」が表示されるだけでは、なにを入力したらいいか、わからないかもしれない。それでは、もう少し親切なプログラムを実行してみよう。

```
10 CLS
20 INPUT "ナマエ　ハ＝"；A$
30 PRINT A$
40 END
```

こんどは「ナマエ　ハ＝」と表示されて、キー入力の状態になるから、なにを入力すればいいか、すぐわかるね。新星太郎なら、「シンセイタロウ」と入力するわけだ。

＊INPUT命令の注意

1．メッセージをつけるときは、" "で囲うこと、そして変数の前に；を忘れないこと。

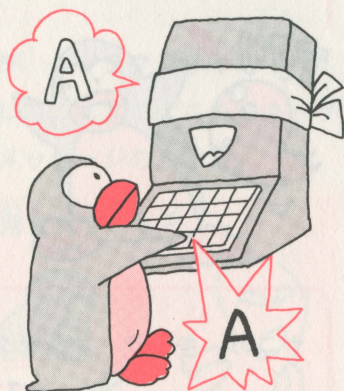
2．文字変数には ,（コンマ）は代入できないよ。, は，「ここでキー入力終わり」という合図になってしまう。

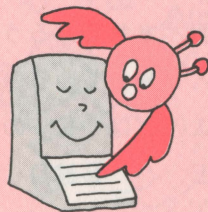
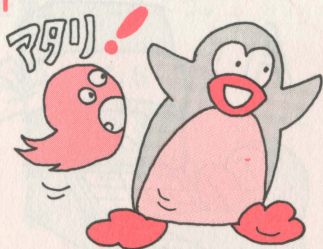
3．キーボードからデータを入力したら、かならず RETURN キーを押す。

♣どんなキーが押されたのか全部わかってしまうINKEY$

キーボードから1文字だけを入力するのがINKEY$命令だ。自分でゲームを作ったりするときに、とてもたくさん使うのが、このINKEY$だから、よくおぼえておこう。

文字変数＝INKEY$

として、この文字変数は押されたキーの文字。そしてINKEY$は、その文字がなにかを調べるというわけだ。

●押されたキーはなにかを調べる

```
10  CLS
20  PRINT "ナニカ キーヲ オシテクダサイ"
30  A$=INKEY$
40  IF A$="" THEN 30
50  IF A$="E" THEN 80
60  PRINT "イマ ";A$;" ヲ オシマシタネ"
70  GOTO 20
80  PRINT A$;
90  PRINT " ハ ENDデス"
100 END
```

ナニカ　キーヲ　オシテクダサイ

イマ　M　ヲ　オシマシタネ

ナニカ　キーヲ　オシテクダサイ

なにも押さないと
いつまでも
「オシテクダサイ」
と出ているよ

プログラムをRUNしよう。

「ナニカ　キーヲ　オシテクダサイ」

と表示されただろう。キーを押してやろうじゃないか。

たとえば「M」を押すと――

「イマ　M　ヲ　オシマシタネ」

と表示されるだろう。どんなキーが押されたか、ファミコンはちゃんと知っているのだ。

こんどは「E」のキーを押してみよう。

「E　ハ　END　デス」

と表示されて、プログラムは終わりだ。

●プログラムの読み方

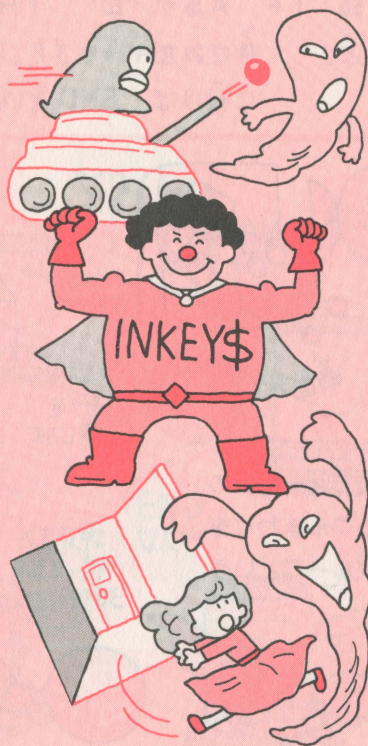
書くほうも読むきみも、少しめんどうくさいが、たまにはプログラムを少しくわしく説明してみよう。

```
10 CLS
20 PRINT "ナニカ キーヲ オシテクダサイ"
```

わかるね。画面をきれいして、まず「ナニカ　キーヲ　オシテクダサイ」というメッセージの表示だ。

＊ INKEY$ はゲームでどんどん利用されている

ファミコンゲームにもいろいろあるね。ロールプレイングゲームからアドベンチャーゲーム、シミュレーションゲームやアクションゲームなど、たくさんの種類が出まわっているけれど、このプログラムの中でも、INKEY$ はたくさん使われているんだ。

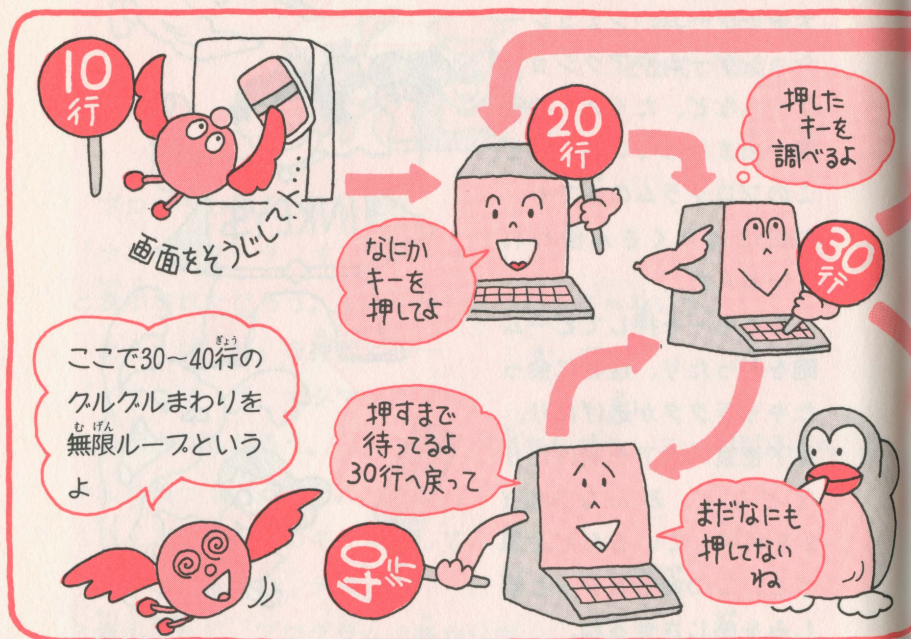
あるキーを押してビーム砲を射ったり、怪物に会ったキャラクタが逃げたり、地下密室のドアを開けたりなどなど、みんな INKEY$ を利用しているんだと思うと、この命令もグッと親しみを感じさせるね。

30 A$=INKEY$

A＄という文字変数、この場合は、押されたキーの文字をA＄に代入している。

40 IF A$="" THEN 30

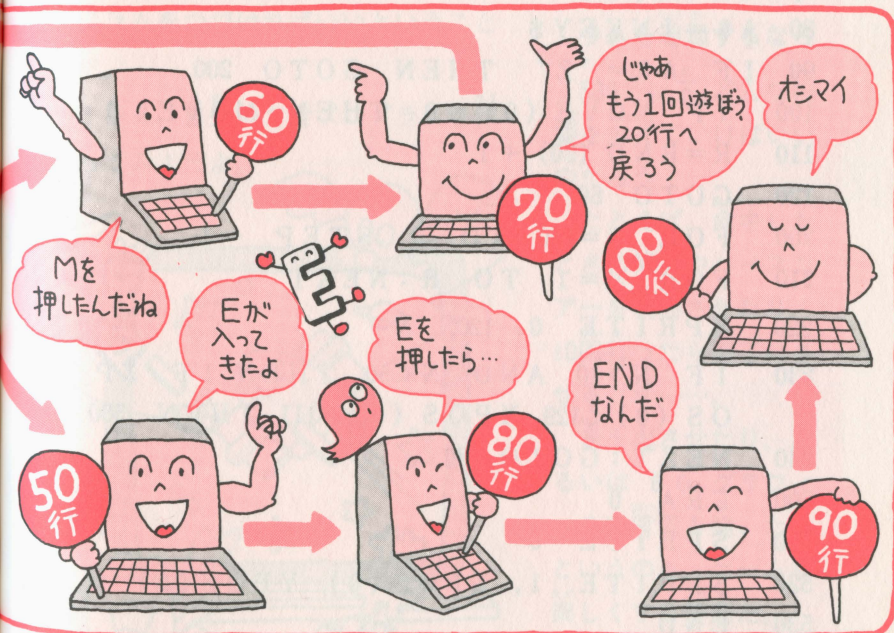
なんのキーも押されていなかったら（A＄=""）、30行に行け。だから、キーが押されないかぎり、30行へ行って40行に行って、また30行に行ってと、グルグルまわりでキリがないよ。

50 IF A$="E" THEN 80

押されたキーが「E」なら、80行へ行け。（そして80行に行けば、ENDということがわかるはずだ。）

```
60 PRINT "イマ "; A$;" ヲ オシマシタネ"
70 GOTO 20
```

Eのキー以外のキーを押した人、たとえば「M」を押したなら「イマ　M　ヲ　オシマシタネ」というメッセージを受け取るはずだ。それでまた20行に戻って最初からやり直し。

```
80 PRINT A$;
90 PRINT " ハ ENDデス"
100 END
```

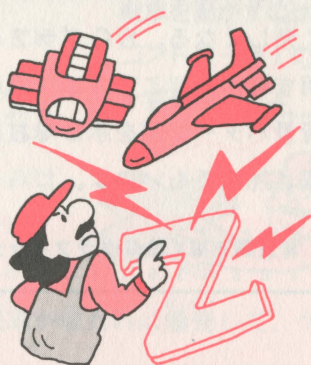
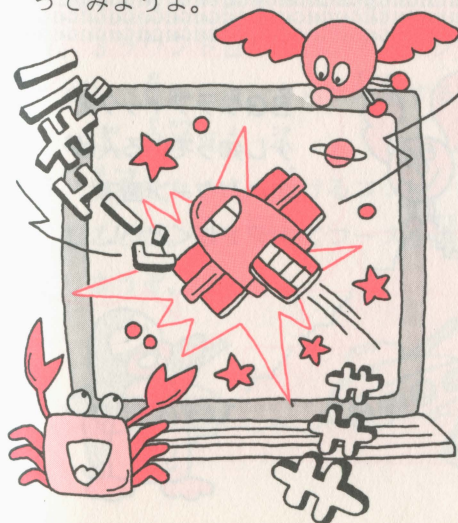
Eキーを押した人には「E　ハ　ENDデス」と表示されて、プログラム終了。

```
10 VIEW
20 SPRITE ON
30 DEF SPRITE 0,(2, 0, 0, 1, 0)=C
   HR$(213)
40 DEF SPRITE 1,(2, 1, 0, 0, 0)=C
   HR$(180)+CHR$(181)+CHR$(182)+C
   HR$(183)
50 DEF MOVE(0)=SPRITE(8, 7, 1,
   25, 0, 0)
60 POSITION 0, 250, 40
70 MOVE 0
80 A$=INKEY$
90 IF A$="Z" THEN GOTO 200
100 IF MOVE(0)=0 THEN MOVE 0
110 R=RND(20)+1
120 GOTO 80
200 FOR N=150 TO 0 STEP -1
210 FOR M=1 TO R:NEXT
220 SPRITE 0, 120, N
230 IF N>40 AND N<50 THEN IF XP
    OS(0)<129 XPOS(0)>111 THEN 500
240 NEXT:GOTO 80
500 ERA 0
510 SPRITE 0
520 SPRITE 1, XPOS(0), YPOS(0)
530 END
```

プログラムを追加してみよう

RUNさせて遊んでみよう。「Z」のキーを押せば、スターキラーを撃墜する弾丸が飛び出すんだ。だけど、少しヘンだと思わない？

スターシップを1回撃墜すると終わってしまうね。このプログラムはまだ、完成していないためなのだ。

1回だけの撃墜で終わらせないで、つづけてしたい人は、530行からプログラムをつづければいいということだ。

さあ、自分の力でプログラムを作ってみようよ。

＊530行以下にプログラムを追加するなら

530行以下、どんな内要を加えていけばいいのだろう？

たとえば、命中したときの得点やハイスコアーを判断させたり、10回打ったら終了にさせるとか……。

音を出させたり、あるいはBGグラフィックで背景を描いたりというのも、ゲームを楽しくする要素だ。

●ゲームがもっと楽しくなるBGグラフィック

背景にＢＧグラフィックで絵を描いておけば、画面がなんとなく楽しくなる。ＢＧグラフィックは、このあとの章でくわしく説明するから、ここでは、ほんの参考程度に、ＢＧグラフィックの背景パターン図を示しておこう。あいているところに、いろいろ描き加えてもいいよ。

■背景画を描くＢＧグラフィックのあるパターン図（スターキラー撃墜作戦）

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1																												
2																												
3																												
4																												
5																												
6																												
7																												
8																												
9																												
10																												
11																												
12																												
13																												
14														G01	G11													
15	G11												G01	G21	G31	G11												G01
16	G31											G01	G21	G21	G31	G31	G11										G01	G21
17	G31	G31	G11								G01	G21	G21	G21	G31	G31	G31	G11								G01	G21	G21
18	G31	G31	G31	G31	G41	G41	G41	G41	G41	G21	G21	G21	G21	G21	G31	G31	G31	G31	G31	G41	G41	G41	G41	G41	G21	G21	G21	G21
19	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41
20	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41	G41

●バトルアタック

ジャーン！

次なるプログラムは、ウォーゲームといこう。BGグラフィックの背景付きだ。

きみは戦車隊の隊長。@マークの味方歩兵が、強力な敵に囲まれて苦戦中。きみは、味方の歩兵を助け出さなければならない。

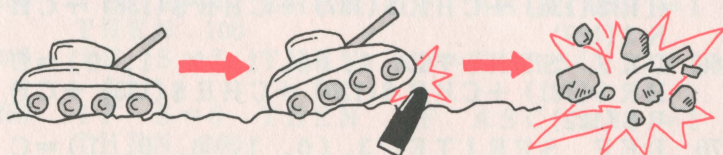
■ゲームのルールと成功のヒント

1．敵の地雷は＊マークだ。これに戦車が触れれば爆発して、ゲームは終了だ。

2．出発点は燃料補給地だ。ここに来ると燃料が補給できる。

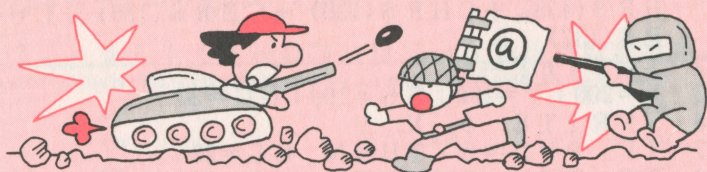
3．@マークを取るたびに、スコアーは10点ずつ加算される。1ラウンドのゲームで、歩兵を10人助け出して燃料基地に戻れば成功で、1ラウンドの終了。再スタートだ。基地に戻ったとき燃料0でもゲームオーバーだ。

4．3ラウンド目からは、＊マークの地雷が少し多くなり、5ラウンド目からは、さらに増加するから、細心の注意を払わなければ、自滅だ。

5．歩兵をたくさん助けるコツは、効率よく燃料を使いながら進むことだ。よく考えてコースを進まないと、燃料がなくなってしまうぞ。

■BGグラフィックのパターン図

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10			
	A10					A10											A10							A10			
	A10					A10											A10							A10			
	A10					A10					A10						A10							A10			
	A10					A10					A10				A10	A10	A10	A10	A10					A10			
5	A10					A10					A10													A10			
	A10			A10	A10	A10	A10	A10			A10													A10			
	A10										A10			A10	A10	A10								A10			
	A10										A10			A10										A10			
	A10													A10						A10				A10	A10	A10	A10
10	A10			A10	A10	A10	A10	A10						A10			A10				A10					A11	A10
	A10								A10					A10			A10			A10						A11	A10
	A10									A10			A10	A10			A10							A10	A10	A10	A10
	A10									A10														A10			
	A10	A10	A10	A10			A10			A10														A10			
15	A10						A10										A10							A10			
	A10						A10										A10	A10	A10	A10				A10			
	A10			A10						A10	A10	A10	A10	A10			A10							A10			
	A10			A10								A10												A10			
	A10			A10								A10												A10			
20	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10			

```
10 CLS
20 CGSET 1,0
30 SPRITE ON
40 VIEW
50 DEF SPRITE 0,(0,1,0,0,0)=CHR$(120)+CHR$(121)+CHR$(122)+CHR$(123)
55 DEF SPRITE 1,(0,1,0,0,0)=CHR$(136)+CHR$(137)+CHR$(138)+CHR$(139)
60 DEF SPRITE 2,(0,1,0,1,0)=CHR$(121)+CHR$(120)+CHR$(123)+CHR$(122)
70 DEF SPRITE 3,(0,1,0,0,1)=C
```

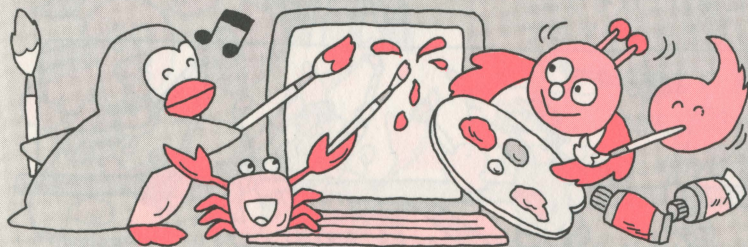
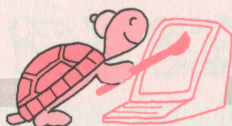
```
HR$(138)+CHR$(139)+CHR$(136)+CH
R$(137)
72 T=0:PP=1
75 X=200:Y=100:SS=100:GG=20
76 SPRITE 0,X,Y
77 IF PP>=3 THEN GG=25
80 IF PP>=5 THEN GG=30
81 FOR N=1 TO GG
82 XX=RND(22)+1
83 YY=RND(19)+1
84 Z$=SCR$(XX,YY)
85 IF ASC(Z$)=1 THEN 82
86 IF N<=14 THEN LOCATE XX,Y
Y:PRINT"@"
87 IF N>=15 THEN LOCATE XX,Y
Y:PRINT"*"
90 NEXT
100 S=STICK(0)
110 IF S=0 THEN 100
120 IF S=1 THEN A=2:B=2
130 IF S=2 THEN A=0:B=0
140 IF S=4 THEN A=3:B=3
150 IF S=8 THEN A=1:B=1
160 IF S=1 THEN A=2
170 X1=X/8-1:Y1=Y/8-2
180 IF S=1 THEN A$=SCR$(X1+1,Y1)
190 IF S=1 THEN IF ASC(A$)=1
THEN 100
200 IF S=2 THEN A$=SCR$(X1-2,
Y1)
210 IF S=2 THEN IF ASC(A$)=1
THEN 100
220 IF S=4 THEN A$=SCR$(X1,Y1
+2)
230 IF S=4 THEN IF ASC(A$)=1
THEN 100
240 IF S=8 THEN A$=SCR$(X1,Y1
-1)
250 IF S=8 THEN IF ASC(A$)=1
THEN 100
```

```
260  IF S=1 THEN X=X+8
270  IF S=2 THEN X=X-8
280  IF S=4 THEN Y=Y+8
290  IF S=8 THEN Y=Y-8
293  SPRITE 0
294  SPRITE 1
295  SPRITE 2
296  SPRITE 3
300  SPRITE A, X, Y
310  SS=SS-1
315  LOCATE 5, 22:PRINT 〝  〟
320  LOCATE 5, 22:PRINT 〝ネンリョウ〟;SS
330  IF SS<=0 THEN 1000
340  IF (X=200 OR X=208) AND (Y=100
     OR Y=108) THEN SS=100
350  V$=SCR$(X/8-1, Y/8-2)
360  IF V$=〝@〟 THEN LOCATE X/
     8-1,Y/8-2:PRINT〝 〟:T=T+10: GOTO 380
363  IF V$=〝*〟 THEN 1000
365  V$=SCR$(X/8-2, Y/8-2)
370  IF V$=〝@〟 THEN LOCATE X/
     8-2, Y/8-2 :PRINT〝 〟 :T=T+10
375  IF V$=〝*〟 THEN 1000
380  LOCATE 15, 22
390  PRINT 〝          〟
400  LOCATE 15, 22
410  PRINT 〝スコアー=〟;T
420  IF (X=200 OR X=208) AND (Y=100
     OR Y=108) THEN IF T>=100*PP TH
     EN CLS:PP=PP+1:VIEW:GOTO 75
430  GOTO 100
1000 LOCATE 10, 22
1010 PRINT 〝GAME OVER〟
```

パート 4

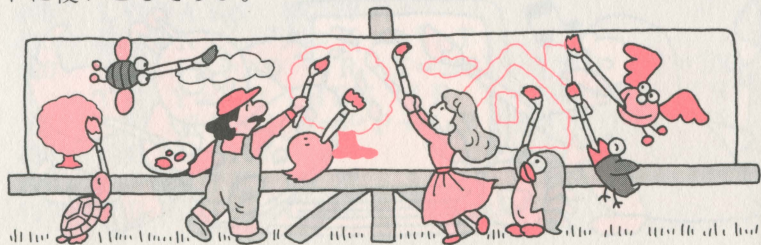
BG グラフィックで絵を描いて

ステップ　1

絵のない ゲームなんて

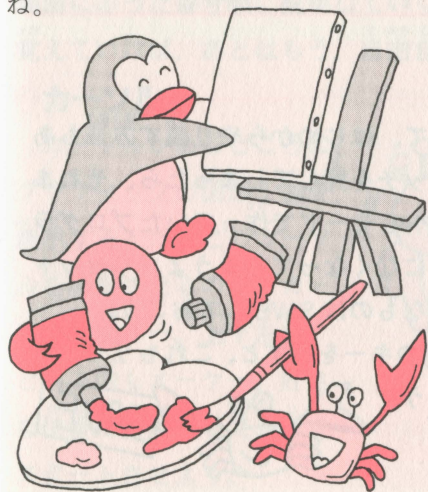
♥ 描いて楽しい、見て楽しい きみだけの背景画を作ろう

ファミコンの画面には、バックグラウンド面とスプライト面の2つがあることは、もう、わかっているよね。

バックグラウンド (Back Ground) とは、背景ということ。そして、ＢＧグラフィックとは、背景画ということだ。

ゲームをするときだって、絵があれば、おもしろさやスリルだって、グーンとアップするし、「迷路ゲーム」のように、ＢＧグラフィックで迷路を作って、その中を通りぬけていくゲームもあるしね。ＢＧグラフィックをおおいに活用して、ファミコンをワイドに使いこなそうよ。

ＢＧグラフィックを起動させよう

ＢＧグラフィックの起動方法には２通りあるから、きみのつごうのいいほうを選んで始めよう。

ひとつは、電源を入れて、GAME BASICモード画面から起動させる方法。これは、BASICを選ぶときに、かならず通るところだから、もう、よく知っているね。

もうひとつは、プログラムを作ったりするBASICの画面から、GAME BASICを呼び出して、ＢＧグラフィックに移る方法だ。この呼び出し命令は「SYSTEM」。覚えておいてね。

＊ＢＧグラフィック

動かない背景を別のところで作って、ゲームそのものとドッキングさせて、より迫力あるゲームや画面にしようというもの。

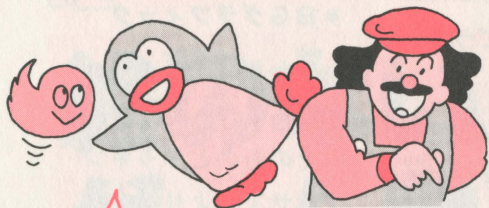
このドッキング命令がVIEW命令だ。

＊VIEW命令

ＢＧグラフィックは、画面上のバックグラウンド面に出力される。ベーシックのＶＩＥＷ命令は、これをバックグラウンド面にコピーする命令だよ。

＊SYSTEM命令

プログラムを作成したりするBASICの画面から、GAME BASICを呼び出すための命令。

■おなじみ、この画面から

BASICにもどりたいなら、ESC キーを押して、次に STOP キー。そうすると、やっぱりこの画面に戻るから、ここで 1 のキーを押せばBASICだ

GAME BASIC

1……BASIC

2……BG GRAPHIC

3……END

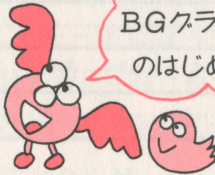
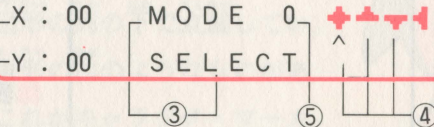
1, 2, 3 KEY IN !!

＊ RESET スイッチに注意

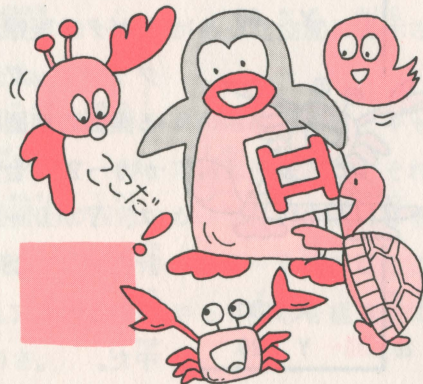
リセットスイッチを押して、はじめからやり直す方法もあるけど、一度リセットスイッチを押してしまったら、それまで BASIC モードや BG GRAPHIC モードで作っていたプログラムや背景画はみんなきれいになくなってしまうよ。リセットスイッチを押す前に、必要なものは SAVE しておこう。

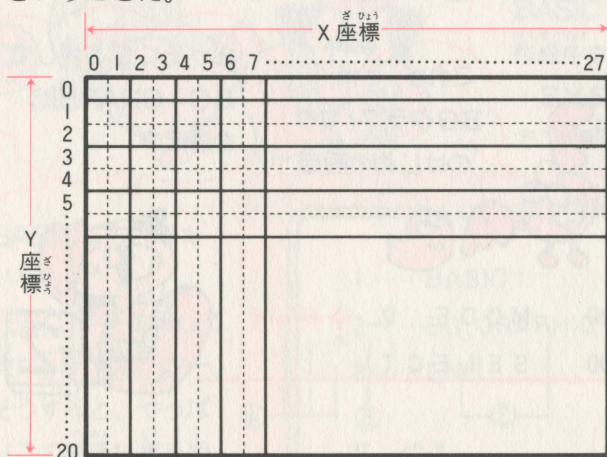
GAME BASIC モードで 3 のキーを押すと、これもリセットされてしまうから注意しよう。

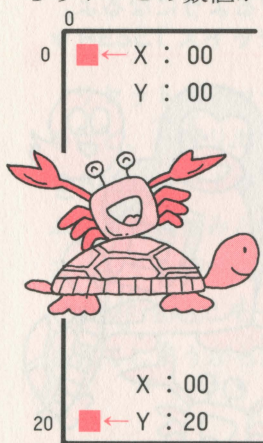
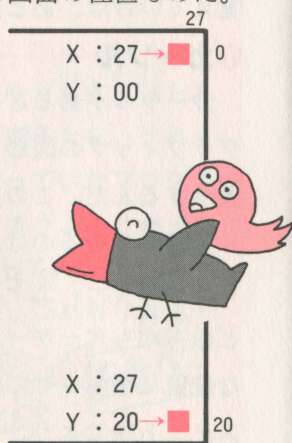
SAVE

♥BGグラフィックの画面(がめん)でどうなっているの？

このＢＧグラフィックの画面(がめん)の使(つか)いかたについて知(し)らないと、実際(じっさい)に絵(え)を描(か)くことができないから、くわしく説明(せつめい)しておこう。画面(がめん)にふった番号順(ばんごうじゅん)に説明(せつめい)していくから、ひとつひとつしっかり覚(おぼ)えていけば、あとはもう、自由自在(じゆうじざい)に描(か)けるようになるよ。

❶カーソル

カーソルがあるところに、グラフィックの図形(ずけい)１個(こ)を表示(ひょうじ)するんだ。このカーソルを移動(いどう)させるのは、カーソル移動(いどう)キー。上下左右(じょうげさゆう)、どの方向(ほうこう)もオーケー。好(す)きな位置(いち)に移動(いどう)させて表示(ひょうじ)をスタートさせよう。

❷座標位置(X，Y)

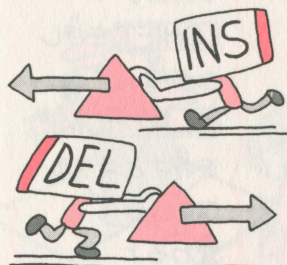
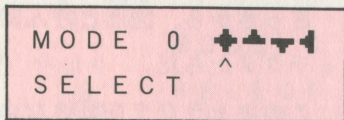
座標なんていうとむずかしそうだけど、表示する場所はどこかということだ。

■表示画面の座標

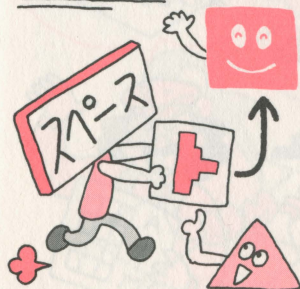
ためしに、カーソル移動キーを押して、カーソルをどこかちがう場所に移してごらん。XとYの右よこに出る数字がかわっただろう？　この数値が、図形１個が表示される画面の位置なのだ。

Xがヨコ座標で、Yがタテ座標。BGグラフィックで描ける画面の範囲は、Xが０～27で合計28マス、Yが０～20で合計21マス、28×21で、合計588のマス目の中だ。

③描くテクニックを示すモード

```
MODE  0  ┿┷┳┫
         ^
SELECT
```

MODE（モード）というのは、図形を画面に表示したり消したりするテクニックの種類のことと考えればいい。

今、SELECTモードになっているね。これは、右に一列に出ている図形をSELECTする（選ぶ）モードということなんだ。

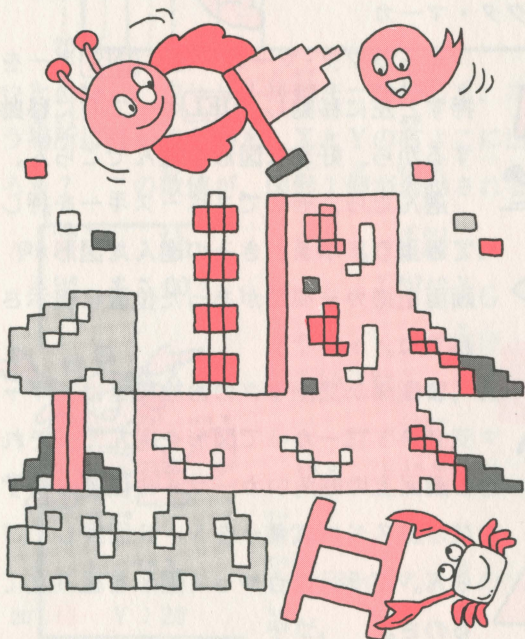
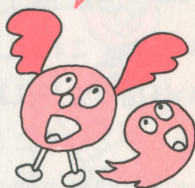
その図形の列の下で点滅している、三角形の山のようなものがあるね。これがキャラクタ・マーカーだ。

キャラクタ・マーカーは、INSキーを押すと左に移動し、DELキーで右に移動するから、好きな図形を選んでごらん。

選んだら、そこでスペースキーを押してみよう。ホラ、きみの選んだ図形が、画面上のカーソルがあった位置に表示されたね。

BGグラフィックのお絵描きは、キャラクタ・マーカーで図形を選んで、それを画面上の四角いカーソルの位置に移すだけ。スペースキーを5～6回押してごらん。ホラ、こうやって図形を組み立てるのさ。

❹ブロック遊びにも似た図形のもと

さっきから、図形と呼んでいるのがこのキャラクタたちだ。

このキャラクタを好きなように組み合わせて、画面というカンバスに絵を描いていくのだが、ナ、ナント、キャラクタはぜんぶで104種類も用意されているのだ。

かなりのデザイン的センス、美的センスが要求されそうだが、これだけあれば、どんなおもしろい背景画だって描けるんじゃないかな？　どんどん、やってみよう。

A～Mまでの13グループにそれぞれ0～7までの8個のキャラクタだ

CLR／HOME を押すとMグループのほうへ

SHIFT を押しながら CLR／HOME を押すとAグループのほうへ変わるよ

⑤マス目４個にひとつの色

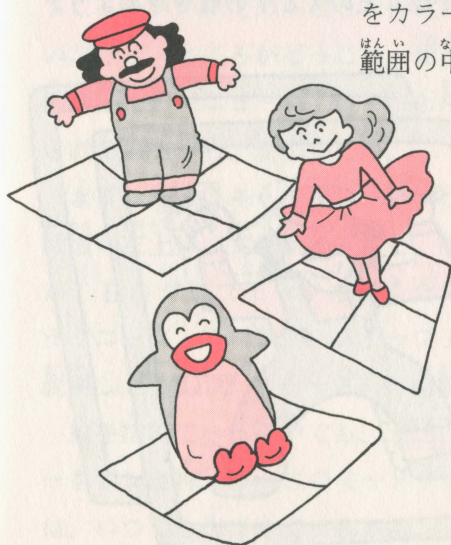
MODE 0
SELECT

MODEのとなりに表示されている「0」は、キャラクタにつける色の番号だ。色は0～3までの合計4色が用意されていて、色を指定するキーは RETURN キーだ。RETURN キーを1回押すと、数字がかわるから、これもためしてみよう。

ところで、この色つけで、とてもたいせつなことがあるんだ。

0 1 2 3 4 5 ……

②で説明した座標位置は、表示される場所だけではなくて、表示される色にもおおいに関係があるんだよ。

ちょっと、左の座標図の一部を見て。色のついた4個のマス目、この4個はぜんぶ同じ色で表示される。この単位をカラーエリアというのだけど、この範囲の中で2色も3色もは使えないよ。

ためしに、この4つのマス目に、それぞれ色を変えてキャラクタを並べていってごらん。先に描いたキャラクタは、あとから描いたキャラクタの色に変わってしまったはずだよ。

ただし、いちばん上の段だけは、はんぱだから、横2つのマス目がカラーエリアだよ。

＊ＢＧグラフィックとアニメキャラクタの合成

スプライト面にはアニメキャラクタを表示し、バックグラウンド面にはＢＧグラフィックを表示するわけだけど、この２つの画面のＸ座標とＹ座標の関係は、少しちがう。

それぞれのＸ座標、Ｙ座標にキャラクタ、または絵を表示して、そのままドッキングさせて表示すると、ぶれてしまうことがあるから要注意だ。

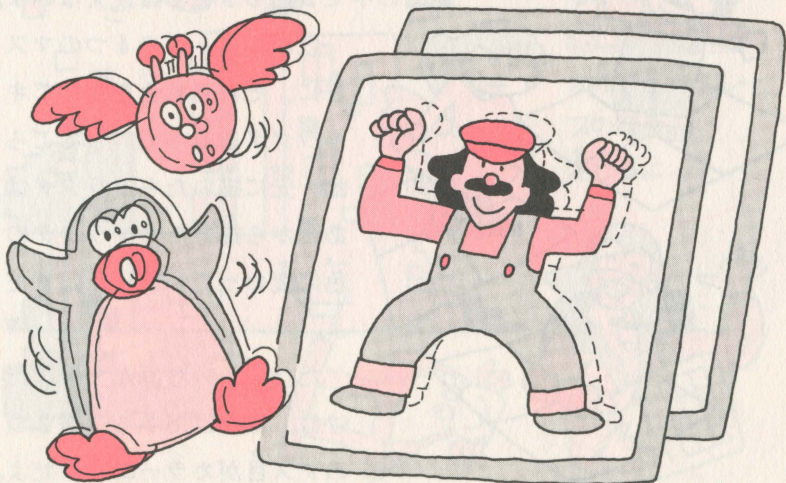
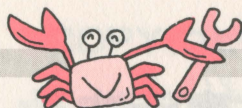
背景の絵にアニメキャラクタを合成させるときの座標は、次の式で割り出すんだよ。

ヨコ

スプライト面(ヨコ)＝(BG GRAPHIC(ヨコ)×８)＋16

タテ

スプライト面(タテ)＝(BG GRAPHIC(タテ)×８)＋16

こうして補正して、プログラム上のＸとＹの値を求めよう。

ステップ　2

テクニックをマスターしよう

♥コマンドをフルに使って きみもテクニシャンになろう

588個のマス目に、ひとつひとつキャラクタを表示させていくなんていうと、すごくたいへんでめんどうな作業みたいだけど、ところがどっこい、便利な描きかたのメニューが、いくつも用意されているから、安心してね。

まず、ESC キーを押してみよう。画面の左上に、文字が出てきた。これが、ＢＧグラフィックを描くためのテクニック・コマンド。ステップ１で説明したMODE（モード）の種類だ。

絵を描いている途中でも、ESC キーをポンと押せば、このモードリストは、いつでも出てきてくれるよ。

```
>SELECT
 COPY
 MOVE
 CLEAR
 FILE
 CHAR
```

ESCキー、ポン、でいつでも出てくるよ

●SELECT（セレクト）

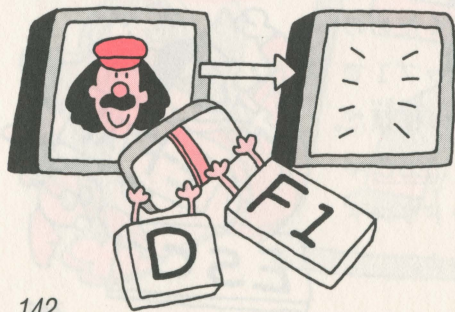
キャラクタや絵を画面に表示する

ＢＧグラフィックを選んだときに、すぐに出てくる画面で使えるのがSELECTだったね。

SELECTには3つの機能があるよ。

1．はじめにいろいろためしたように、数多くのキャラクタの中から選んで画面に表示させる。

2．これから説明する、他のいろいろな機能を使って、描いたり消したり、くふうしたりして作った画面を、ひとつのまとまった画面として表示させる。

3．キャラクタを消す。アルファベットのＤキーか、ファンクションのＦ１キーを押すと、カーソルの中のキャラクタが消える。

＊消すときは色に注意

SELECTでキャラクタを消すとき、Ｆ１キーを使うと、同時に配色も変えてしまうから、注意してね。

●COPY（コピー）

キャラクタをいくつもコピーする

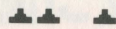
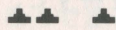
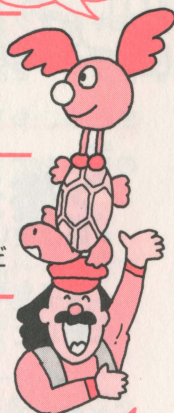
お次は「COPY」を選んでみよう。画面下のMODE表示の下は「COPY」となった。

COPYはコピーだよ。SELECTで同じキャラクタを1個1個表示させてもいいけど、もっと簡単な方法はないかな。というわけで、同じキャラクタをいくつもコピーしちゃうのだ。

いったんセレクトモードで描き込んだキャラクタの中で、いくつも使いたいキャラクタがあったら、その上にカーソルを重ねて INS キーを押す。

そのキャラクタを描き込みたい位置にカーソルを移動させて、DEL キーを押す。

また移動させて DEL キーを押す。またまた移動させて……いくつも同じキャラクタが描けたね。ね、便利だろう。これならラクチンチン、だ。

これはね、INS キーを押したときに、そのキャラクタがカーソルにコピーされて、カーソルがスタンプのような役わりにかわったからなんだ。スタンプなら、なん回押しても、同じものが出てくるわけさ。

●MOVE(ムーブ)

キャラクタの引っ越しだ

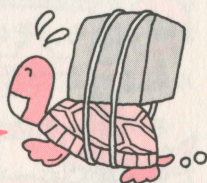
「MOVE」は、「動かす」という意味。カーソルを重ねたところのキャラクタをそっくり別の位置に移動させるんだ。

COPYは、もとのキャラクタがそのままもとの位置に残るけど、MOVEは動かしちゃうわけだから、もとの場所にキャラクタは残らない。キャラクタを別の位置に置いてきたら、もちろん、カーソルの中みもカラッポになる。

キーの役わりが変わるぞ

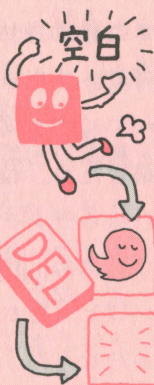
オイショとINSキー

DELキーでお引っ越し

カーソルはスタンプから、引っ越し屋になるわけだ

＊頭は使いようだぞ

なにも表示されていないところにカーソルを移動してINSキーを押すと、空白のコピーだ。

このカーソルを消したいキャラタの上に重ねてDELキーを押すと——そう、空白のスタンプを押しちゃったのだから、そのキャラクタは消えちゃう。

いちいちSELECTにもどって消さなくてもいい、COPYを使った修正テクニックだ。

●CLEAR(クリア)

消してもいいのかをよく考えて

マーカーを「CLEAR」にあわせて、スペースキーを押したとたん、アッ――、絵がぜんぶ消えてしまった。もう一度、同じ絵を出すにはどうしたらいいのか？　かわいそうだけど、オシマイ。

このCLEARで画面を消してしまうと、再生することはできないのだ。うっかり使うと取り返しのつかないことになるよ。

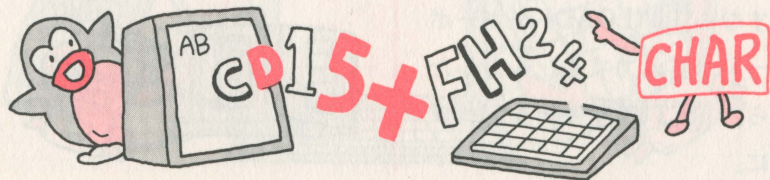
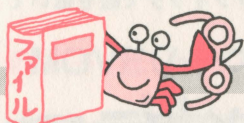
一部分だけ消したいときは、SELECTに切りかえて、Dキーを使って消したほうが安全だ。

●CHAR(キャラクタ)

グラフィック画面に文字を書く

SELECTやCOPY、MOVEで絵を描いているときには、キーボードにあるふつうの文字や数字、記号などは画面に描き込むことはできないのだけど、CHARならできるんだ。

直接、キーボードから打ち込めるから、いろいろためしてごらん。

ステップ　3

描いた絵はとっておこうね

♥テープに記録しておけばゲームともすぐにドッキング

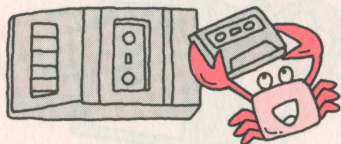
画面に描いた絵は、いったん電源を切るとあとかたもなく消えてなくなってしまうし、新しい絵を描きたいときは、どうしても画面をきれいにしなければならないし、あとでゲームプログラムとドッキングさせたいから、今描いた絵は残しておきたいし。

まかせなさい、「FILE」です。

●FILE(ファイル)

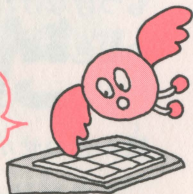
カセットテープをデータレコーダにセットする。ESC キーでメニューを出したら、FILE を選んでスペースキーだ。

テープに記録するなら S (SAVE)、テープから呼び出すなら L (LOAD) をキーボードから入力する。入力したら、RETURN キーを忘れずに。

SAVE (S), LOAD (L)? ■

次にファイルネームを聞いてくるから、16文字以内でつけて入力だ。

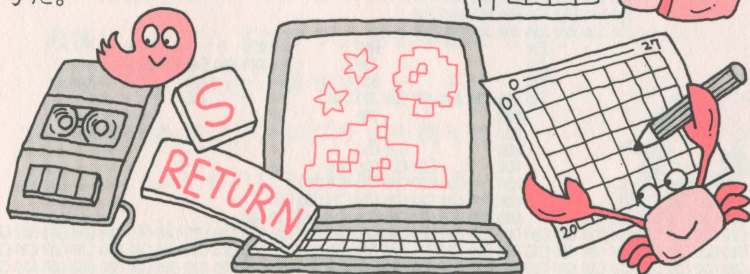
絵にあったタイトルをつけておくといいね。「メイロ」とか「ヤマ」とか……。

```
SAVE (S), LOAD (L) ? S■
FILE NAME ″■
```

RETURNキーで記録開始

♥ 座標もキャラクタ番号も色もまるごと紙に記録しちゃおう

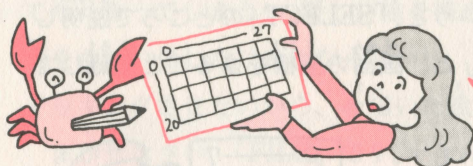
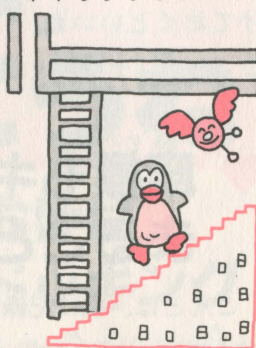
こんどは、紙に記録しておく方法だ。こうしてとっておけば、いつでも、なん回でも利用できるよ。

なにを記録しておくかというと、「SELECT」のところで説明した、「キャラクタテーブルB」と、画面の座標をまるごと記録しておくんだ。

この表のタテにふられているアルファベットがキャラクタグループ。ヨコにふられている数字が番号だ。

たとえば、「Ｆのグループの７番目」というように記録していくんだよ。そして、色もつけなければならないから、０～３までの配色番号を加える。

すると、こんなアルファベットと数字の組み合わせができるだろう。

これがひとつのキャラクタを表わすのだから、同じようにして、表示したキャラクタと色を記録していけばいい。

記録するときには、まず紙にちゃんと画面の座標を作ること。入力するときに便利だし、ひと目でどんな絵かわかるだろう？

あまりうまくないけど
サンプルを作ってみたから
ためしに画面に表示してみてね

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1																												
2																												
3																												
4						I60	J30	J30	J30	J30	J30	J30	I70															
5						J20	F40	F50	F50	F50	F50	F60	J20															
6						J00	J30	J30	J30	J30	J30	J30	J10															
7									J20						H40				H70				H70					
8									J20						J20				E40	E40	E40	E40	E70					
9									J20						J20						J20							
10									J20	J30	J30	J30	J30	J30	J50	J30					J30							
11										H20					J50													
12										H20																		
13										H20				G00	G10													
14										H20			G00	G20	G30	G10												
15				G00	G10					H20		G00	G20	G20	G30	G30	G10											
16			G00	G20	G30	G10				H20	G00	G20	G20	G20	G30	G30	G30	G10										
17		G00	G20	G20	G30	G30	G10			G00	G20	G20	G20	G20	G30	G30	G30	G30	G10									
18	G20	G20	G20	G20	G30	G30	G30	G30	G20	G20	G20	G20	G20	G20	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30
19	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30
20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G20	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30	G30

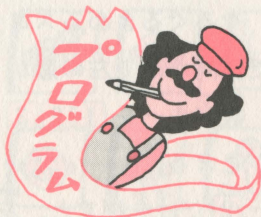
それ行けゲームだ *Let's Key in!*

◆ 遊んだあとはオリジナル負けないゲームを作ろうよ

ひととおりファミリーベーシックの勉強も終わった。あとはきみの努力しだいだ。わからなくなったら、へこたれないでなん回でも読み返そうね。でも、本ばかりにらんでいたって、プログラミングは上達しないよ。どんどん key in することだ。

最後に、ゲームプログラムを2本載せておこう。

まず、正確に key　in すること。そして、遊んだあとで、プログラムを読んでみるといいよ。それほどむずかしいプログラムじゃないから、次は、これらに負けないプログラム作りに挑戦だ。

ゲームプログラム　1

スロットキャラクタ

◆一発勝負のスロットマシン　ファミコンでギャンブルだ

マリオやレディ、ペンペンやニタニタなど、ファミコンのスターたちがスロットマシンに顔を出す。スリル満点のファミコンギャンブルに挑戦してもらおう。

●得点キャラもあれば減点キャラもある

スロットマシンの3つの窓に顔を出すのは、マリオ、レディ、アキレス、ペンペン、ファイターフライ、ニタニタのおなじみのスターたちだ。

マリオ、レディ、アキレス、ペンペンは、きみたちももう知っているように、右向き、左向き、歩いていたりと、6つのスタイルがあるけど、どのスタイルでもそろえばオーケーだ。

ファイターフライとニタニタは2つのスタイルで出てくるけど、この2つは減点キャラクタだから、あまり出てほしくないね。

●遊びかた

ぐるぐるまわるスロットマシンをとめるのは、3つの数字キーだ。向かって左の窓が1、まん中が2、右はしの窓が3だ。

数字キーを 1 2 3 と順に押していくと、それぞれ順番に回転がとまる。

2 だけを押すと、左の窓とまん中の窓がとまる。3 を押すと、

3つの窓の回転がぜんぶ一度にとまるよ。

●ゲームのルール

1．持ち点以上はかけられない。また、小数点の点数もかけられない。

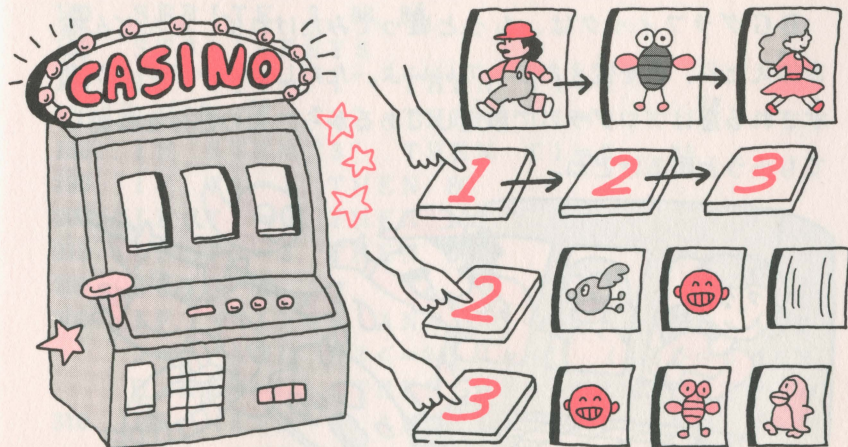
2．3つの窓に同じキャラクタがそろったらかけ点の10倍の得点になる。

3．3つの窓がぜんぶそろわないときは、かけ点分だけ減点になる。

4．左から2つそろうと、かけ点の5倍の得点になる。

5．ファイターフライかニタニタが、1と2の窓に2つそろってしまうと、かけ点の10倍の減点になる。

6．ファイターフライかニタニタが、1の窓に出たら、かけ点の5倍の減点になる。

7．ニタニタがまん中の窓に出ると、かけ点の5倍の減点になる。

けっこうシビアなスロットマシンだけど、それだけにおもしろいはずだよ。減点キャラのニタニタなんかが、ニタニタ～なんて顔を出してたら、ホントに頭にきてしまって、ますますアツくなっちゃうね。

●プログラムとBGグラフィックについて

ゲームを動かすためのプログラムは10行から410行。1000行から1120行、2000行から2030行は画面に文字を表示するためのプログラムだ。もちろん、続けてkey inして、行番号の小さい順から実行されていくわけだけど、こういうプログラムの書きかたも覚えておくと便利だよ。

独立したブロックは、行番号をとばして、たとえば1000行からとか2000行とかきりのいい行番号にして組み込んでおくと、ひと目でそのブロックのプログラムがなんのためのプログラムかがわかるからね。もちろん、メインルーチンの飛び先などには注意してね。

BGグラフィックは、もっと凝ってみたければ、どんどん描き加えたり、色を変えたりしていいよ。ただし、キャラクタが表示される窓はプログラムで設定してあるから、窓の部分を移動してもらうと困るけどね。

■ゲームプログラム――スロットキャラクタ

```
10   CLS
20   CGSET 1, 0
30   SPRITE ON:VIEW
40   S=100
50   GOSUB 1000
60   A1=RND(26)*4
70   DEF SPRITE 0,(0,1,1,0,0)=CHR$(A
     1)+CHR$(A1+1)+CHR$(A1+2)+CHR$(A
     1+3)
80   A2=RND(26)*4
90   DEF SPRITE 1,(0,1,1,0,0)=CHR$(A
     2)+CHR$(A2+1)+CHR$(A2+2)+CHR$(A
     2+3)
100  A3=RND(26)*4
110  DEF SPRITE 2,(0,1,1,0,0)=CHR$(A
     3)+CHR$(A3+1)+CHR$(A3+2)+CHR$(A
     3+3)
120  SPRITE 0,40,80
130  SPRITE 1,64,80
140  SPRITE 2,88,80
150  ST$=INKEY$
160  IF ST$="1" THEN T1=1
170  IF ST$="2" THEN T1=2
180  IF ST$="3" THEN T1=3
190  IF T1=1 THEN 80
200  IF T1=2 THEN 100
210  IF T1=3 THEN 300
220  GOTO 60
300  IF (A1>=0 AND A1<=24) AND (A2
     >=0 AND A2<=24) AND (A3>=0 AN
     D A3<=24) THEN S=S+C*10:GOTO 50
310  IF (A1>=0 AND A1<=24) AND (A2
     >=0 AND A2<=24) THEN S=S+C*5:
```

```
    GOTO 50
320 IF (A1>=28 AND A1<=52) AND (A2
    >=28 AND A2<=52) AND (A3>=28 AN
    D A3<=52) THEN S=S+C*10:GOTO 50
330 IF (A1>=28 AND A1<=52) AND (A2
    >=28 AND A2<=52) THEN S=S+C*5:
    GOTO 50
340 IF (A1>=64 AND A1<=84) AND (A2
    >=64 AND A2<=84) AND (A3>=64 AN
    D A3<=84) THEN S=S+C*10:GOTO 50
350 IF (A1>=64 AND A1<=84) AND (A2
    >=64 AND A2<=84) THEN S=S+C*5:
    GOTO 50
360 IF (A1>=96 AND A1<=108) AND (A2
    >=96 AND A2<=108) AND (A3>=96 AN
    D A3<=108) THEN S=S+C*10:GOTO 50
370 IF (A1>=96 AND A1<=108) AND (A2
    >=96 AND A2<=108) THEN S=S+C*5:
    GOTO 50
380 IF(A1>=56 AND A1<=60)AND(A2>=56
    AND A2<=60)THEN S=S-C*10:GOTO 50
385 IF (A1>=88 AND A1<=92) AND (A2>
    =88 AND A2<=92) THEN S=S-C*10:G
    OTO 50
390 IF (A1>=56 AND A1<=60) OR (A1>
    =88 AND A1<=92) THEN S=S-C*5:G
    OTO 50
395 IF (A2>=88 AND A2<=92) THEN S=
    S-C*5:GOTO 50
397 IF (A1>=56 AND A1<=60) AND (A2
    >=88 AND A2<=92) THEN S=S-C*10:
    GOTO 50
400 S=S-C:T1=0
410 GOTO 50
```

```
1000  LOCATE 18, 3：PRINT"        "
1010  LOCATE 13, 3
1020  PRINT"モチ コイン ハ"; S
1030  IF S<=0 THEN GOTO 2000
1040  LOCATE 19, 10
1050  PRINT"コイン ヲ"
1060  LOCATE 20, 11
1070  PRINT"カケテ クダサイ"
1080  LOCATE 22, 13
1090  INPUT C
1100  IF C>S THEN 1000
1110  T1=0
1120  RETURN
2000  CLS
2010  LOCATE 13, 3
2020  PRINT"GAME OVER"
2030  END
```

スロットキャラクタのBGグラフィック

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																	
1																	
2																	
3			D60	D60	D60	D60	D60	D60	D60	D60	D60	D60					
4		D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60				
5		D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60				
6		D60	L32	L62	L62	L62	L62	L62	L62	L62	L62	L72	D60		G70		
7		D60	L42			L42			L42			L42	D60		C30		
8		D60	L42			L42			L42			L42	D60		C30		
9		D60	L52	L62	L62	L62	L62	L62	L62	L62	L62	M02	D60		C30		
10		D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60		C30		
11		D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	K50	K10		
12		D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60	D60				
13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13			
14	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13			
15	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13	H13			
16																	
17																	
18																	
19																	
20																	

ゲームプログラム　2

マスターマインド

◆ 4つの数の組み合わせ きみのカンがためされるぞ

遊びかたはかんたんだ。ファミコンがかくし持っている4つの数をあてるゲームだ。AからDまでの4つの箱に、コレカナ？と思った数を入れていくだけ。1から9までのうち4つの数字を組み合わせると……これはもう、目がまわるくらいあるぞ。

■ゲームプログラム――――マスターマインド

```
10    CLS
20    CGSET 1, 0
30    VIEW
40    DIM A (8)
50    E=1
60    FOR I=0 TO 8
70    K=RND (9)
80    A (I) =K
90    IF K=0 THEN 70
100   IF I=0 THEN 140
110   FOR J=0 TO I-1
120   IF A (J) =K THEN 70
130   NEXT
140   NEXT
200   IF A (0) =0 OR A (1) =0 THEN 50
210   IF A (2) =0 OR A (3) =0 THEN 50
220   A=A (0)
```

```
230  B=A(1)
240  C=A(2)
250  D=A(3)
260  LOCATE 9,22
270  INPUT "A=";A1
280  LOCATE 4,7
290  PRINT A1
300  LOCATE 9,22
310  INPUT "B=";A2
330  LOCATE 9,7
340  PRINT A2
350  LOCATE 9,22
360  INPUT "C=";A3
370  LOCATE 14,7
380  PRINT A3
390  LOCATE 9,22
400  INPUT "D=";A4
410  LOCATE 19,7
420  PRINT A4
430  LOCATE 9,22
440  PRINT "      "
450  IF A=A1 AND B=A2 AND C=A3 AN
     D D=A4 THEN 500
460  LOCATE 14,16
470  PRINT E;" カイメ デス"
480  E=E+1
490  GOTO 260
500  LOCATE 4,22
510  PRINT" オ オ ア タ リ"
520  LOCATE 5,22
530  PRINT"モウイチドスルトキハ"
540  LOCATE 8,23
550  PRINT" RUNヲシテクダサイ。"
560  END
```

マスターマインドのＢＧグラフィック

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0																								
1																								
2																								
3																								
4		D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41
5		D41																						D41
6		D41		L30	L60	L60	L70		L30	L60	L60	L70		L30	L60	L60	L70		L30	L60	L60	L70		D41
7		D41		L40			L40		L40			L40		L40			L40		L40			L40		D41
8		D41		L50	L60	L60	M00		L50	L60	L60	M00		L50	L60	L60	M00		L50	L60	L60	M00		D41
9		D41																						D41
10		D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41	D41
11							D43	D43	D43	D43	D43	D43	D43	D43	D43	D43	D43	D43						
12							D43	D43	D43	D43	D43	D43	D43	D43	D43	D43	D43	D43						
13												D40	D40											
14												D40	D40											
15												D40	D40											
16												D40	D40											
17												D40	D40											
18				D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40			
19			D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40		
20		D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	D40	

★ファミリーコンピュータ・ファミコン、
ファミリーベーシックは任天堂の商標です。

この本の内容についての問い合わせは、
株式会社　新星出版社　☎03-831-0743
ファミリーコンピュータ係までお願い
いたします。

ファミリーコンピュータ™
絵でわかるファミコンベーシック

1986年8月15日　初版発行　©

著　者　山　下　利　秋
発行者　富　永　弘　一
印刷所　慶昌堂印刷株式会社

発行所　東京都台東区 台東2丁目24　株式会社　新星出版社
郵便番号110　電話(831)0743　振替東京4-72233

ISBN4-405-06072-X

ファミリーコンピュータ™&ファミリーベーシック™

ぼくらのファミコン入門

ファミリーベーシックを覚えてゲーム作り。キーボードの操作法とベーシックによるプログラミングの基礎をわかりやすく説明した。

B6判
定価780円

絵でわかる ファミコンベーシック

新星出版社

定価680円

●ファミリーコンピュータ・ファミコン、ファミリーベーシックは任天堂の商標です

ISBN4-405-06072-X C2055 ¥680E

ファミリーコンピュータ™ ファミリーベーシック™

絵でわかる ファミコンベーシック

新星出版社

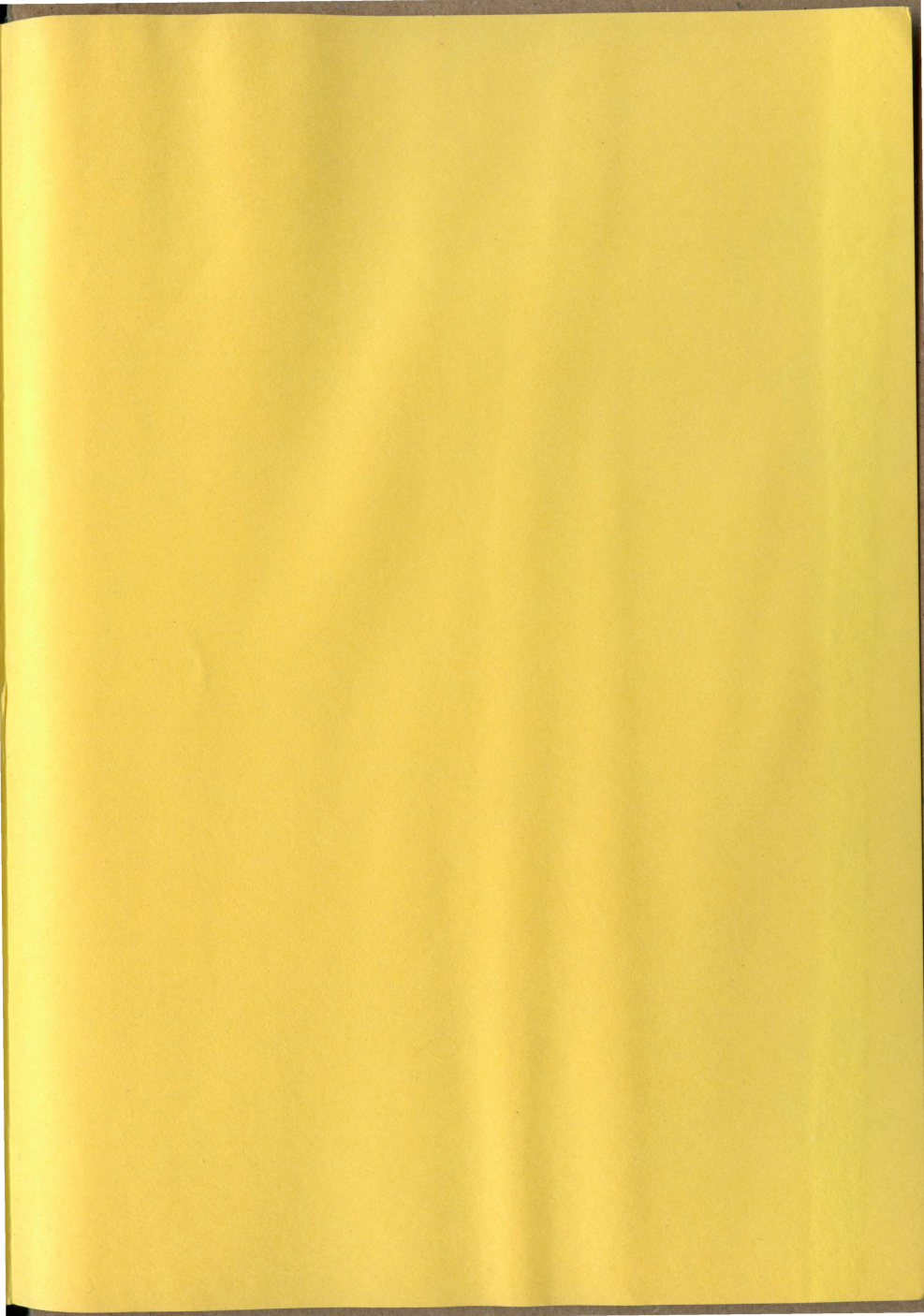
385

ファミリー コンピュータ™
ファミリー ベーシック™

絵でわかる
ファミコンベーシック

山下利秋

新星出版社

ファミリーコンピュータTM
ファミリーベーシックTM

絵でわかるファミコンベーシック

新星出版社
385

絵でわかる
ファミコンベーシック

新星出版社

●ファミリーコンピュータ・ファミコン、
ファミリーベーシックは任天堂の商標です